RL-TR-96-246
Final Technical Report
February 1997

VIRTUAL PATH MANAGEMENT FOR SURVIVABLE ATM NETWORKS

Carnegie Mellon University

Hyong S. Kim

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-96-246 has been reviewed and is approved for publication.

APPROVED: Richard C. Butler II
RICHARD C. BUTLER II
Program Engineer

FOR THE COMMANDER: John A. Graniero
JOHN A. GRANIERO
Chief Scientist
Command, Control & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/C3BC, 525 Brooks Rd, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1997	FINAL SEP 94 - MAR 96

4. TITLE AND SUBTITLE

VIRTUAL PATH MANAGEMENT FOR SURVIVABLE ATM NETWORKS

5. FUNDING NUMBERS

C - F30602-94-C-0236
PE - 62702F
PR - 4519
TA - 22
WU - 31

6. AUTHOR(S)

Hyong S. Kim

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Carnegie Mellon University
Department of Electrical and Computer Engineering
Pittsburg, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory/C3BC
525 Brooks Road
Rome, NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-96-246

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Richard C. Butler, II/C3BC, (315) 330-7751

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Fast restoration from a network failure has been recognized as a key ingredient in realizing survivable networks in emerging high-speed ATM environments. Self-healing techniques have been proposed to provide service continuity to end-users by autonomously switching affected virtual paths to alternate routes. However, their success greatly depends on how traffic is distributed and spare capacity is dimensioned over the network when a failure occurs. Therefore, a fast restoration mechanism alone is not enough to achieve a high level of survivability. This project aims at defining an integrated framework for survivable ATM network management ranging from network design and virtual path management to fast restoration protocol. The focus is placed on two aspects essential to promoting restoration capability: the survivable virtual path routing (SVPR) and the survivable capacity and flow assignment (SCFA).

14. SUBJECT TERMS

Asychronous Transfer Mode (ATM)
High-Speed Networking
Survivability
Network Management

15. NUMBER OF PAGES

192

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Table of Contents

List of Figures

List of Tables

Abstract

Fast restoration from a network failure has been recognized as a key ingredient in realizing survivable networks in emerging high-speed ATM environments. Self-healing techniques have been proposed to provide service continuity to end-users by autonomously switching affected virtual paths to alternate routes. However, their success greatly depends on how traffic is distributed and spare capacity is dimensioned over the network when a failure occurs. Therefore, a fast restoration mechanism alone is not enough to achieve a high level of survivability. This project aims at defining an integrated framework for survivable ATM network management ranging from network design and virtual path management to fast restoration protocol. The focus is placed on two aspects essential to promoting restoration capability: the survivable virtual path routing (SVPR) and the survivable capacity and flow assignment (SCFA). The SVPR attempts to maximize restorability from a possible network failure by adjusting virtual path configuration in response to a change in traffic demand or facility network configuration. The two-step restoration concept is also introduced to achieve fast restoration as well as optimal reconfiguration. The SCFA seeks for the most economical physical link capacity placement in order to deploy a survivable network in a cost-effective manner. Joint optimization is carried out to find a globally optimal solution. The problems for maximum-flow restoration-based networks are formulated as a large-scale linear programming model. Several mechanisms are devised to reduce computation time by facilitating the basis matrix factorization. The SVPR for k-shortest path restoration-based networks is modeled as a non-smooth multicommodity flow problem with linear constraints. A modified flow deviation method is developed to obtain a near-optimal solution in a reasonable computation time. Premature convergence to a non-smooth point is avoided by properly adjusting optimization parameters. Using the proposed optimization procedures, the minimum resource installation cost is quantitatively analyzed for different fast restoration schemes. Contrary to a wide belief in the economic advantage of the end-to-end restoration scheme, this study reveals that the attainable gain could be marginal for a well-connected and/or unbalanced network.

I Introduction

1.1. Problem Statement

Survivability has become a critical issue in telecommunication networks due to increasing societal dependence on communication systems and the growing importance of information. The significance has been further magnified by the advent of high-capacity optical fiber: a 2.4 Gbps optical transmission system over a single wavelength is already available, with an increased rate beyond 10 Gbps projected for the near future [56]. Using the wavelength division multiplexing techniques, many channels of different wavelengths can be multiplexed on a single optical fiber, and each optical trunk is expected to support a data rate ranging from a hundred Gbps to Tbps [34] [56] [66]. In order to utilize the high transmission capacity of optical fiber cables and reduce the transmission cost, the topology of wide-area networks is becoming sparse, and data streams are aggregated into relatively few optical fiber cables [55] [56]. A large concentration of transmission capacity in the exchange networks would considerably increase the vulnerability of the network and reduce the robustness of the network to a transmission component failure. In this environment, a network failure such as an optical link impairment or a node failure can cause a large loss of data, even in a short outage. Thus, it is imperative to make the service interruption time as short as possible, and a fast restoration from a network failure has become one of the key survivability issues for future high-speed networks.

In order to mitigate the impact of a network failure, self-healing techniques have been developed for mesh-type long-haul or metropolitan exchange networks based on Synchronous Transfer Mode (STM) [2] [35] [80] [82] as well as Asynchronous Transfer Mode (ATM) technologies [5] [43] (see Section 2.1.2 for details). Their main goal is to provide service continuity so that a network failure is almost imperceptible to end-users. For this purpose, fast restoration schemes[1] autonomously recover affected traffic in a short period after a failure so that the existing connections would not get dropped[2]. In ATM networks, virtual paths[3] passing over an impaired network component are rerouted upon failure. A fast virtual path (VP) restoration scheme dispatches the spare

1. The terms "self-healing" and "fast restoration" are used interchangeably.

2. The restoration should finish before the outage duration reaches call-dropping threshold or before data protocol timeout happens [35]. Most existing services become impacted by a 2 to 10 second service outage [75]. Thus, fast restoration algorithms typically aim at recovering affected transport paths in 2 seconds [36] [80] [83].

3. A virtual path (VP) is a semi-permanent logical transport path where multiple calls (connections) are accommodated (see Section 2.1.1 for details).

bandwidth required for VP rerouting and switches the affected VP's over the discovered alternate routes.

The primary concern in previous works was the development of a fast restoration protocol which accelerates the restoration procedures (restoration speed) and recovers affected bandwidth as much as possible (restoration ratio or restorability). The proposed algorithms have been evaluated with respect to these two measures through computer simulation. A few sample networks are employed in the evaluation where link flow and spare bandwidth assignments are predetermined.

With dynamically changing traffic demand, however, flow assignment varies over time, and the amount of link spare capacity changes accordingly. But flow assignment and spare bandwidth allocation have a direct influence on the restorability of a fast restoration scheme because the flow assignment determines the amount of bandwidth to be recovered when a network failure occurs, and spare bandwidth is utilized for the restoration. Thus, in order to offer better network survivability, it is crucial for a network manager to realize restorable flow assignment in response to a change in traffic demand as well as facility network configuration.

Physical link capacity allocation also has a significant influence on the restorability because it determines a set of feasible flow and spare bandwidth assignments for a given traffic demand. Full restorability against major network failures could be guaranteed by installing a sufficiently large amount of fiber links throughout the network. However, installation costs could easily rise, especially in a public inter-office network which covers a geographically wide area. Therefore, a proper network design procedure is essential in deploying a survivable network in a cost-effective manner.

This project addresses these two open issues for survivable ATM networks with self-healing capabilities. The interrelationship of these survivability issues is illustrated in Figure 1-1. Given an available physical network resource and a traffic demand, the *survivable virtual path routing (SVPR)* algorithm finds a virtual path routing and bandwidth assignment which minimizes an unrecoverable amount of flow due to possible network failures. This problem arises when there is a change in the traffic demand patterns or in the underlying facility network. The *Dynamic virtual path reconfiguration* algorithm is performed on such occasions to maximize the restorability of the fast restoration procedure under the new network environment. This function offers a better sur-

vivability under a changing network environment, especially for a multi-rate service system like an ATM network where gross traffic demand patterns could be susceptible to a change in a relatively small number of high volume traffic. Furthermore, a fully restorable network can be realized with less resource installation in a dynamically reconfigurable network compared to its static counterpart. Since physical link capacity cannot be increased in real-time, static VP configuration requires overengineering in order to cope with varying demand patterns [29].

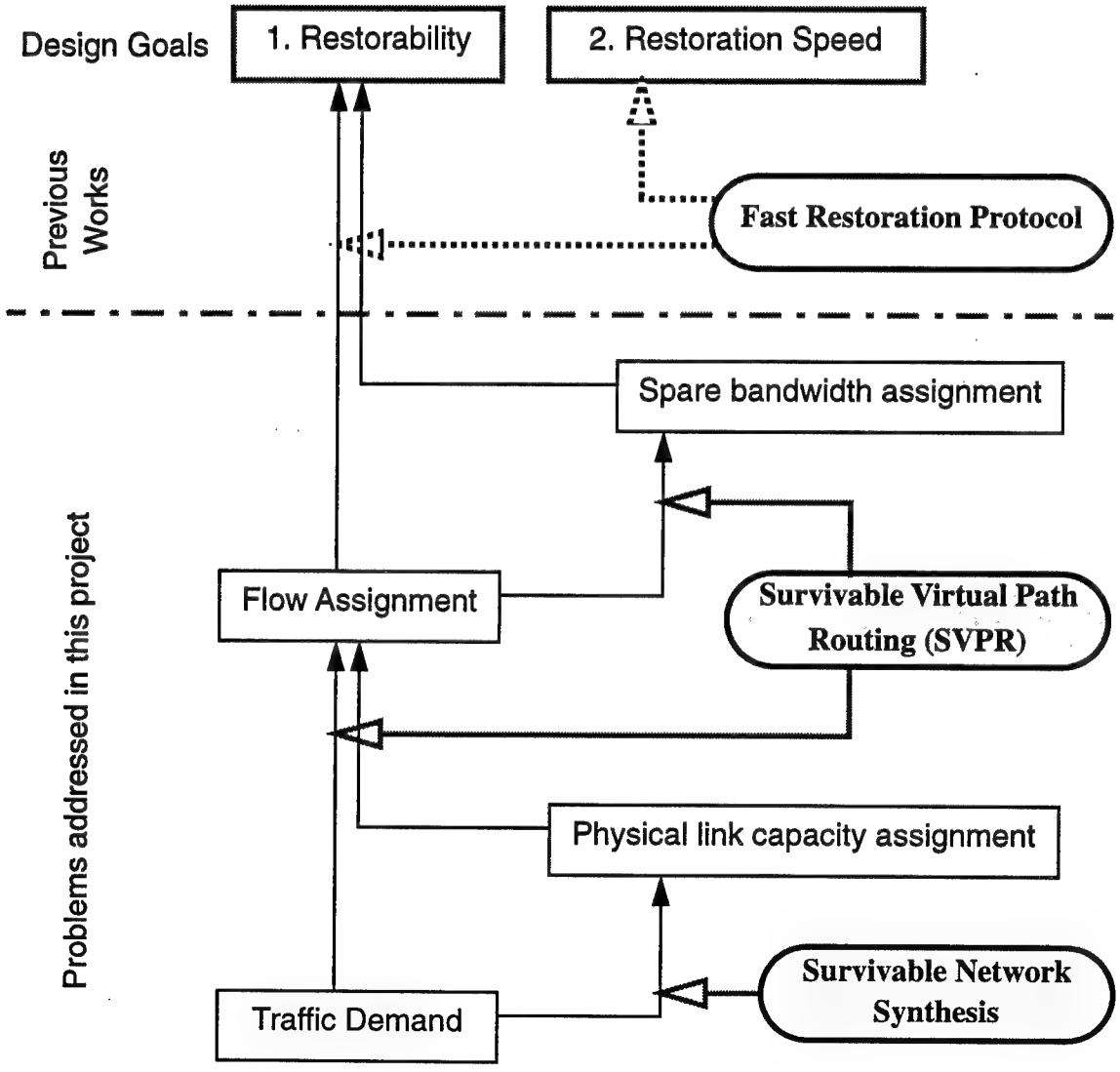


Figure 1-1. Survivable ATM network management issues

Given a network topology and a projected traffic demand, the *survivable network synthesis* algorithm seeks for the most economical link capacity assignment that assures full restorability against any major network failure. This design process and the construction of an additional facility are initiated whenever the dynamic VP reconfiguration process cannot maintain survivability at a desired level due to a growth of traffic demand over months or years.

1.2. Previous Work

Three related research fields and their previous development are reviewed in this section, including survivable/reliable network management in conventional networks, fast restoration schemes, and dynamic virtual path management in ATM networks. Virtually all works have been developed in separate contexts, and none have consolidated all three elements.

Network synthesis and flow assignment problems have been widely discussed in the context of traditional telephone networks as well as conventional store-and-forward data networks, although most works aim at designing networks under normal operating conditions. Relatively fewer works exist for survivable/reliable network management, where traditional network reliability measures for low-speed networks are accommodated in their problem formulation. Depending on the reliability criterion, these works can be roughly categorized into multiple connectivity problems and non-simultaneous flow assignment problems.

The *multiple connectivity approach* has been used extensively in traditional reliability literatures [3] [25]. It ensures the existence of multiple diverse connections between any communicating pair. Recently, the idea has been applied to reliable virtual path management [53]. The availability of virtual paths is guaranteed even with a transmission link failure by distributing bandwidth over at least two VP's of every switching pair. As for a network synthesis problem, the multiple connectivity approach uses global availability[4] as a network reliability measure and designs a network with a minimum cost subject to the upper bound of network unavailability. This approach would be appropriate for traditional store-and-forward networks where multiple connectivity prevents the connection from being broken and allows rerouting. Network congestion and packet overflow could be readily avoided by reactively suppressing the flux of packets into the network. Although this would incur additional delay and temporal degradation of quality of service, large data loss

4. Global availability is defined as the probability that the network is simply connected.

can be prevented. Lost packets can also be recovered by a small number of packet retransmissions. On the other hand, this reactive flow control is inappropriate for high-speed networks since a large amount of cells are under transmission over giga-bit links. Thus, huge data loss is unavoidable before reactive action can be initiated due to relatively slow feedback [10].

The connectivity-based reliability measure also poses a serious problem for traditional low-speed networks. The approach overlooks the level of performance experienced by users in the presence of a failure [25]. A significant number of queuing delays or lost calls would result unless sufficient bandwidth is provided. Moreover, a connectivity measure would be inappropriate for a conventional telecommunication network because it is typically well-connected even after a network failure [70]. The *non-simultaneous flow assignment approach* has been proposed to remedy the above pitfalls. The approach chooses a set of failure events which are most likely to happen and solves multiple flow assignment problems corresponding to each failure scenario [70]. Expected lost calls can be employed as a reliability measure instead of connectivity. Recently, the idea has been applied to SONET-based transport networks [29] as well as ATM VP-based transport networks [30]. They assure full service restorability for any failure pattern in the predefined scenarios, and a new call will be rejected if there is no way to allocate it without violating this full restorability requirement. This means that call level quality of service (QOS) may be degraded during a busy period in order to achieve 100% restorability against failures, which should be a relatively rare event. The network synthesis problem based on the non-simultaneous flow approach minimizes the resource installation cost while assuring the existence of a feasible flow assignment in any failure state [41] [58] [59]. The problem can be formulated as a non-simultaneous multi-commodity flow problem. The resulting network can accommodate the traffic demand projected at the planning phase even in the presence of any failure scenario considered in the design.

A major weakness of the above approaches is their failure to model a lost flow upon a network failure. They assume it can be neglected as a temporal event. Although the obtained flow assignment can satisfy traffic demand at a steady state even if a failure is present, it could incur critical service interruption when a system switches to another steady state upon failure. This could be especially problematic in high-speed wide-area networks based on ATM technologies with each link operating at the order of Gbps to Tbps. Not only a very precise synchronization of path switching is necessary over the network, but also the cell sequence of virtual channels in all VP's must be preserved. Since a very large number of ATM cells are under transmission over gigabit or

terabit transmission links, even a small difference in the path length between old and new routes entails a large delay gap of the cells traversing over the routes. This huge gap must be absorbed somehow upon path switching for all existing VP's across the network, which is almost impractical. Path switching without this precaution will cause severe service degradation all over the network, affecting even the connections which do not suffer directly from the failure. A huge number of dropped cells must be retransmitted for all VP's at once, which would significantly overload the entire network and might cause long-lasting total service degradation. On the other hand, the self-healing algorithm reroutes only affected VP's at the time of a failure. Although temporal service interruption is still inevitable due to the delay from the time of failure to the activation of alternate VP's, it is mostly confined to the VP's directly affected by the failure. Thus, the service degradation would disappear more quickly. Finally, the network designed by the non-simultaneous flow assignment approach might not be able to recover all affected flow since a self-healing algorithm usually requires extra capacity to fulfill the run-time restoration. In order to circumvent the above issues, a fast restoration protocol must be modeled in the problem formulation, although virtually all previous literature on survivable network synthesis and survivable flow assignment have failed to meet this requirement.

Fast restoration schemes have been extensively discussed[5] since Grover introduced the idea of self-healing networks [35]. However, none have addressed the survivable flow assignment problem in this context. Only a few papers have discussed the capacity design problem for an STM-based self-healing network [37] [68] [69] [78]. Assuming a predetermined link flow assignment, they seek a spare link placement with minimum capacity installation cost, subject to the constraint that fast restoration would succeed under any single link failure. The obtained solution, however, is not necessarily optimal for a given projected traffic demand because a different flow assignment may produce a more economical capacity assignment. Instead, joint optimization of link capacity and virtual path flow assignment must be carried out in order to find a truly optimal solution [24].

In the previous works on self-healing schemes, the main focus has been placed on the restoration speed and the restoration ratio of the restoration protocols. The effectiveness of the proposed algorithms has been demonstrated with respect to these two measures. A few sample networks have been employed in these experiments, where a physical link placement and a link flow assignment are given. However, the performance of a fast restoration protocol is largely determined by physi-

5. Refer to Section 2.1.2 for references and details.

cal network resource and flow allocation as discussed in the previous section. Thus, the evaluation of the fast restoration schemes on an arbitrarily assigned network resource might not produce meaningful results, although such an approach has been often taken in the literature. This is mainly because previous works lack an integrated view of survivable network management from network design and flow assignment to fast restoration protocol. The current literature also lacks comparative study of different restoration protocols through quantitative measures. Although several major alternatives exist for self-healing protocols which significantly differ in their implementation strategies (see Section 2.1.2 for details), their comparison has been performed only qualitatively or through simulation on a sample network.

Dynamic virtual path (VP) management has become a hot topic in the field of ATM traffic management [27] [47] [51] [62] [64] [65] [71] [72] [73]. The introduction of virtual path concepts significantly reduces the burden of ATM traffic control due to its ability to handle multiple virtual channels as a bundle (see Section 2.1.1 for details), but it comes at the expense of resource utilization. By adaptively updating virtual path routing and bandwidth assignment, a dynamic VP management scheme attempts to enhance the network resource utilization under a quality of service (QOS) objective, such as cell loss rate and call blocking probability. However, the vast majority of the previous works were conducted under an implicit assumption of a 'fail-free' network. Only very recently has the issue of survivable dynamic virtual path management been addressed in the literature [30], but fast restoration has not been modeled in their approach.

1.3. Objectives and Organization

This project aims to define a complete framework of the survivable network management ranging from a physical network resource design and a dynamic virtual path reconfiguration to fast restoration[6]. This is the first work to incorporate fast restoration control in ATM resource management. The relevant issues have been previously addressed without any mutual consideration of other important elements. As shown in Figure 1-2, the proposed network management system integrates them in order to realize efficient and cost-effective ATM network management with restoration capabilities. The fast restoration capability is especially important for public long-haul

6. The term "survivability" has been used in various contexts in the literature. For example, it was considered to be 'the capability of a network where a certain percentage of the traffic can still be carried immediately after a failure' [57]. In this project, "survivability" is defined as 'the ability to provide service continuity upon network failure'. In the ATM network environment, it can be interpreted as the restorability of the fast VP restoration scheme.

or metropolitan exchange networks. Since a large amount of traffic is aggregated, an impact of a network failure is extremely large in such networks. This research considers the ATM inter-office networks as its primary application area.

The focus is placed on the survivable network synthesis problem and the survivable virtual path routing (SVPR) problem. The key objectives of this project are 1) to develop the optimization models which take into account fast restoration schemes and 2) to propose and implement the solution procedures for the models in order to improve ATM network survivability. Three representative fast restoration protocols are considered (see Section 2.1.2 for details), and they are independently modeled in the formulation of the survivable network synthesis as well as SVPR problems. The accommodation of a fast restoration protocol considerably enlarges the size of the problem. Thus, the challenge is to develop algorithms which make problems computationally trac-

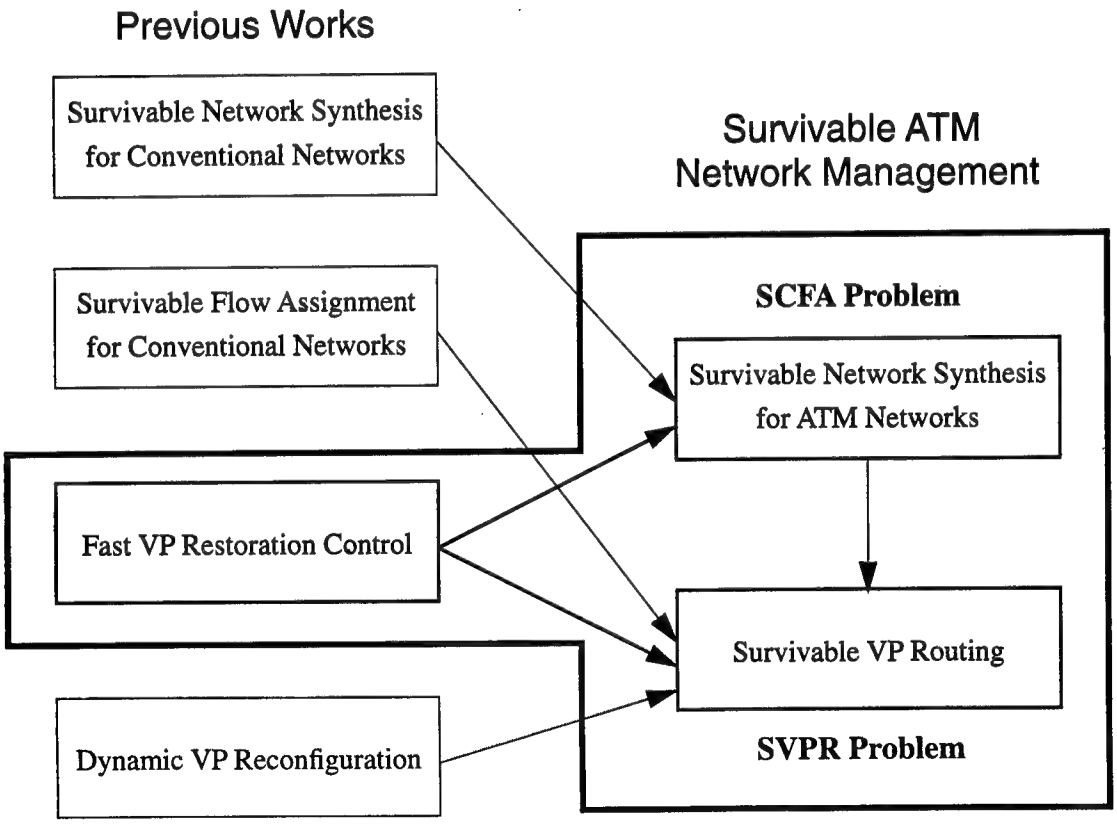


Figure 1-2. An integrated view of survivable ATM network management

table for networks of a realistic size. *Survivable ATM network management architecture* is also introduced to simplify the complexity of survivable network management problems.

As for the network synthesis problem, the joint optimization of capacity and flow assignment is carried out to find a globally optimal solution for a given link cost function[7]. Since there is mutual dependency between the capacity placement and the survivable flow assignment, the obtained solution cannot be claimed to be optimal if these problems are treated separately. We call this problem the *Survivable Capacity and Flow Assignment (SCFA)* problem. The problem formulation differs by restoration schemes, and distinct solution procedures are developed accordingly. In this design, full restorability against any single link failure is assured for a projected traffic demand[8].

The SVPR problem aims to achieve a higher level of survivability by minimizing the expected amount of lost flow upon fast restoration from a link failure. A VP configuration and bandwidth assignment is reconfigured in response to a dynamic change of network environment so that a self-healing algorithm would succeed. Since the amount of lost flow is calculated based on the self-healing algorithm, the solution gives an optimal traffic distribution with minimum service interruption. We also introduce a concept of *two-step restoration* which accommodates two contradicting requirements after a failure: fast restoration and optimal VP reconfiguration.

The proposed optimization procedures give us a tool for comparative study among different fast restoration schemes. As mentioned before, their pros and cons have been only discussed qualitatively or based on simulation results on a certain sample network. With the SCFA solution procedures, restoration schemes can be quantitatively compared with respect to the optimal resource installation cost. This comparative analysis helps to clarify the economical benefit of the restoration schemes. The comparison is performed using several networks with diverse topological characteristics as well as various projected traffic demand patterns. The SVPR solution procedures further elucidate the advantage of the restoration protocols in terms of their attainable restorability. An interesting result contrary to a widely believed hypothesis is drawn from this analysis.

This report is organized as follows. The next section elaborates the proposed survivable ATM network management system and introduces the survivable ATM network management architec-

7. The word 'optimality' is used throughout the report in a sense that an obtained solution could minimize or maximize a given objective function.

8. Thus, this design aims to construct a cost-effective one-fault-tolerant network.

ture and the two-step restoration concept. The SCFA problem is discussed in Section III, including the problem formulations as well as the development of solution procedures. Comprehensive analysis of the economic benefit of each restoration protocol is also explored. Section IV is devoted to the SVPR problem, the problem formulations and their solution procedures. The effectiveness of the proposed dynamic VP configuration schemes as well as the two-step restoration scheme is demonstrated through numerical experiments. These experiments employ the link capacity assignment obtained through the SCFA solution procedure for a given projected traffic demand. The performance of the proposed SVPR procedures is examined at various network loads around the projected demand arising in a practical situation. Finally, Section V summarizes the major contributions of this research, and the report concludes with possible extensions for future research.

II A Survivable ATM Network Management System

ATM network resource management requires highly complicated procedures since resource allocation requests from several levels of traffic entities (i.e. ATM cells, calls and virtual paths) must be handled effectively to meet various types of quality of service (QOS) objectives (e.g. cell loss rate and call blocking rate). A layered management architecture can simplify the network management process by classifying different types of network resources and traffic entities into layers [40]. However, the existing architecture is developed without any consideration of network survivability. This work proposes a survivable ATM network management architecture, where survivability functions are embedded into the virtual path layer and the facility network layer. Dynamic network reconfiguration is performed at these two layers in order to enhance network survivability. A two-step restoration strategy is further proposed to meet two contradicting requirements of virtual path restoration and is implemented at the virtual path layer. A survivability measure is defined to assess the actual amount of loss the network would suffer in the event of a failure. This measure is employed as a decision criterion for resource allocation control in a survivable ATM network management system.

2.1. Background

This section reviews the technologies relevant to survivable ATM network management. They include the ATM and virtual path concepts and the ATM resource management scheme. ATM resource management has been widely discussed in the literature, but all previous works lack the consideration of network survivability. This section also surveys a variety of fast restoration schemes in detail.

2.1.1. ATM and virtual path concepts

Asynchronous Transfer Mode (ATM) has been adopted by CCITT as a transfer mode for Broadband Integrated Service Digital Network (BISDN) [1]. The ATM is a switching and multiplexing technique based on a short, fixed-size packet called a cell [61]. Unifying the packet structure enables fast switching and allows multi-media data with diverse traffic characteristics to be sup-

ported in an integrated manner. Figure 2-1 depicts the ATM cell format standardized by CCITT.

A logical transport path, called a Virtual Path (VP), has been proposed for the ATM networks [71] [72] and adopted by CCITT [1]. A VP is a logical connection for a pair of nodes which are not necessarily linked by a single physical cable. As shown in Figure 2-2, virtual paths create a virtual path sub-network over a facility network. Multiple virtual channels (individual service connections) are accommodated in a VP, and they are processed as a bundle. Each VP is assigned a route and a bandwidth that determines the upper limit of virtual channels contained in the path. A virtual path is specified by the Virtual Path Identifier (VPI) within a header, while a virtual channel is

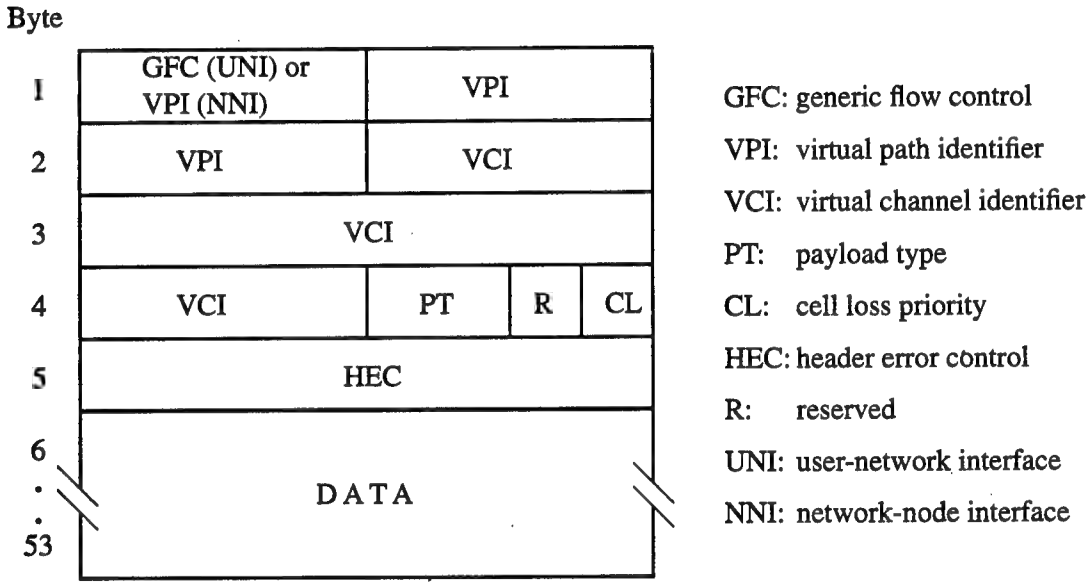
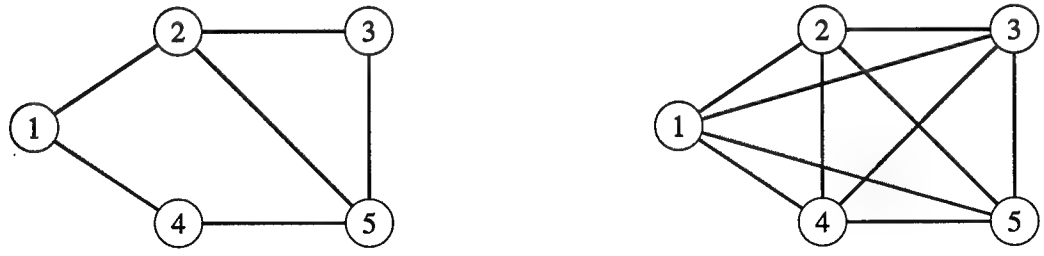


Figure 2-1. The ATM cell structure



(a) Facility (physical) network

(b) VP sub-network

Figure 2-2. A virtual path sub-network

identified by the combination of the VPI and the Virtual Channel Identifier (VCI) (Figure 2-1).

Figure 2-3 illustrates an example of a VP-based ATM transport system. Direct virtual paths are established from Node 1 to Node 3 through Node 2 (VP 1) as well as from Node 1 to Node 2 (VP 2). Each virtual path is shared by multiple virtual channels. Two types of ATM switches are introduced: an ATM virtual path (AVP) switch and an ATM virtual channel (AVC) switch. An AVP switch is directly connected to transport links at each node, while an AVC switch is installed between an AVP switch and subscriber lines. An AVP switch checks only the VPI field of incoming cells. A cell is relayed to an outgoing trunk if the node is a transit node for the VP the cell belongs to (e.g. Node 2 for VP 1 in the above example). A cell is sent to an AVC switch only if the VP is terminated at the node (e.g. VP 2 at Node 2). The AVC identifies the service entity for the cell based on the VPI and VCI values and directs the cell to end-user equipment or to another VP.

Major advantages of a virtual path are simplified network management, increased flexibility in resource management and enhanced network reliability. First of all, the task of a call set-up is simplified by the introduction of virtual paths. A new call is accepted or rejected depending on the current usage of a virtual path bandwidth. If accepted, a part of VP bandwidth is reserved for this call. This call admission and bandwidth allocation process can be performed only at the source

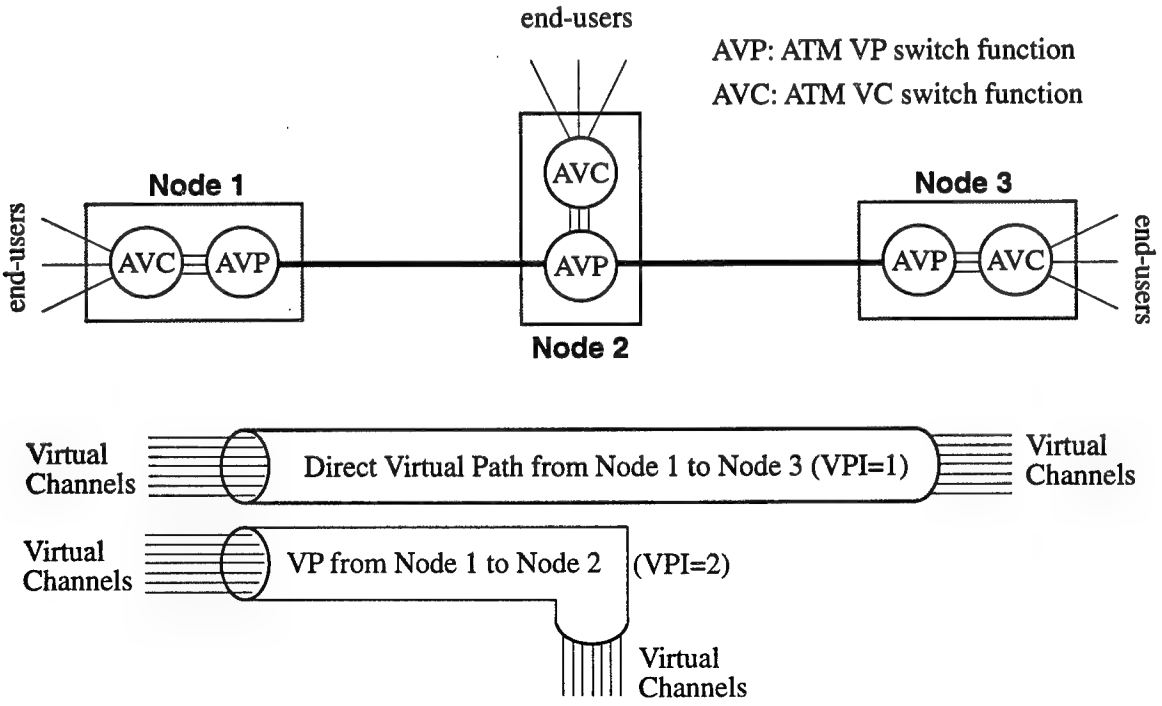


Figure 2-3. A VP-based ATM transport system

node of a VP. Furthermore, the routing tables must be updated only at the end nodes of a virtual path. Thus, the transit nodes of a VP are free from call-by-call based processing during a call set-up time [71]. The processing load of a transit node is further reduced by grouping virtual channels into a VP. Cells are relayed to an outgoing link only by checking their VPI field at an AVP switch. Without a VP, each node must hold switching information on all existing connections in a look-up table, including the ones relayed at the node.

Dynamic VP reconfiguration capability gives flexibility in network resource management and realizes a high utilization of transmission resources. The VP routing and bandwidth allocation can be updated in response to a long-term fluctuation or growth of the traffic demand. If a VP bandwidth must be fixed, it should be equal to the maximum required capacity in the long run. This static VP configuration is very inefficient since a link capacity cannot effectively be allocated to virtual paths under varying traffic demand patterns [65]. The best resource utilization can be achieved by optimizing the VP routing and bandwidth assignment over a physical network for a given traffic demand. A dynamic VP reconfiguration approach can also adapt to a topological change in the physical network, such as a new facility installation or a change due to a network failure.

Finally, and most importantly for survivable network management, virtual paths can be rerouted in case of failure. Since VP restoration requires fewer operations than rerouting each call individually, it is expected to restore more services in a shorter period [82].

2.1.2. Fast path restoration

Extensive research has been conducted on fast path restoration schemes in mesh-type long-haul or metropolitan exchange networks since Grover [35] introduced an idea of self-healing networks in 1987 [11] [15] [18] [36] [45] [68] [79] [80] [82] [83]. Originally, the self-healing network was designed for a synchronized path (e.g. the DS-3 path) restoration in STM networks. An automatic synchronized path switching system called a Digital Cross Connect (DCC) switch is utilized to switch affected paths over alternate routes upon a network failure. Recently, the same technology has been extended to ATM networks where restoration path switching is carried out at a virtual path level [5] [23] [43] [52]. Due to fast ATM cross connect switching, ATM virtual path restoration has the potential of achieving rapid restoration [82]. Furthermore, the spare capacity can be allocated logically over working trunks in a VP-based restoration system. Thus, the spare bandwidth can also help alleviate temporal traffic congestion.

When a network failure happens, the fast path restoration mechanism recovers affected transport paths by rerouting them over alternate paths using spare bandwidth. A restoration protocol typically takes the following four phases:

Phase 1. Failure detection and notification

Phase 2. Restoration path hunting

Phase 3. Restoration bandwidth reservation

Phase 4. Path rerouting

The following is one of the typical scenarios employed in self-healing networks [82] [83]: When a link failure happens (Phase 1), one end of the failed facility is designated a Sender node. A Sender broadcasts restoration messages toward the other end of the link, which is called a Chooser node (Phase 2). All nodes other than a Sender and a Chooser work as a tandem node. A hop count limit is imposed at this message flooding phase in order to prevent unnecessary message circulation. A tandem node writes its node ID and available spare bandwidth information into a message, and the node relays the updated message to its adjacent nodes. The routing information is accumulated as a message traverses. When a message reaches the Chooser node, the stored information constitutes an available restoration route between the two end nodes. Upon arrival of the message, the Chooser node sends an acknowledgment message back to the Sender over the discovered alternate route. The spare capacity is reserved at this phase (Phase 3). Finally, the Sender issues a confirmation message back to the Chooser which subsequently triggers the transport path switching over the route (Phase 4). This procedure usually results in finding a set of successive shortest paths and reserving their maximum spare bandwidth until enough restoration capacity is found. A double-search algorithm is also proposed to perform bidirectional restoration [23]. Two Sender nodes and two Chooser nodes are defined, and the above procedures proceed in parallel from the two end nodes of a failed link to find restoration paths in both directions.

Various other implementation techniques have been also proposed. Fast restoration algorithms can be classified as follows by their path rerouting strategies [19] [82]:

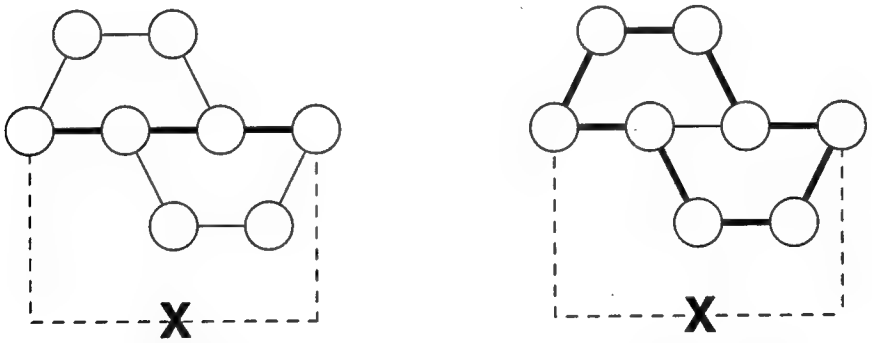
1. Restoration path selection: k-shortest path-based versus maximum flow-based.
2. Network segment to restore: line restoration versus end-to-end restoration[1].

The self-healing algorithm discussed above employs k-shortest path-based line restoration.

1. Some literature refers to line restoration as link restoration or local rerouting. End-to-end restoration is also referred to as path restoration, point-to-point restoration, or source-based rerouting.

Realizing faster restoration and restoring more affected flow are two major goals in self-healing networks, which lead to two alternative restoration path selection strategies: *k*-shortest path (KSP) based restoration and maximum flow (MF) based restoration [20] [82] (Figure 2-4). The KSP-based restoration aims for quick recovery of affected traffic. As soon as a restoration path is found in Phase 2, the KSP restoration protocol reserves the available bandwidth over the path in Phase 3 and reroutes (a part of) affected demands over this newly discovered alternate path in Phase 4. This operation is repeated in parallel until all affected bandwidths are restored or the restoration process is terminated due to a lack of available spare bandwidth. In other words, the KSP restoration algorithm recovers interrupted traffic at the earliest possible time, whenever any part of the restoration paths become available. However, this procedure may not be able to find the maximum available spare capacity [79] (see Figure 2-4). The MF-based restoration approach, on the other hand, aims to find a rerouting pattern which allows maximum bandwidth to be restored. For example, Whalen et.al. [79] propose to employ a variant of the maximum flow algorithm.

Although the MF algorithm can detect more restoration bandwidth, it is computationally complex and very hard to meet the 2-second restoration time objective. Due to the substantial delay, it is imperative that the MF-based fast restoration system precalculates restoration routes and plans rerouting patterns beforehand [11]. The information on the restoration routes is distributed and stored at relevant nodes. When a failure occurs, affected traffic is rerouted using this information.



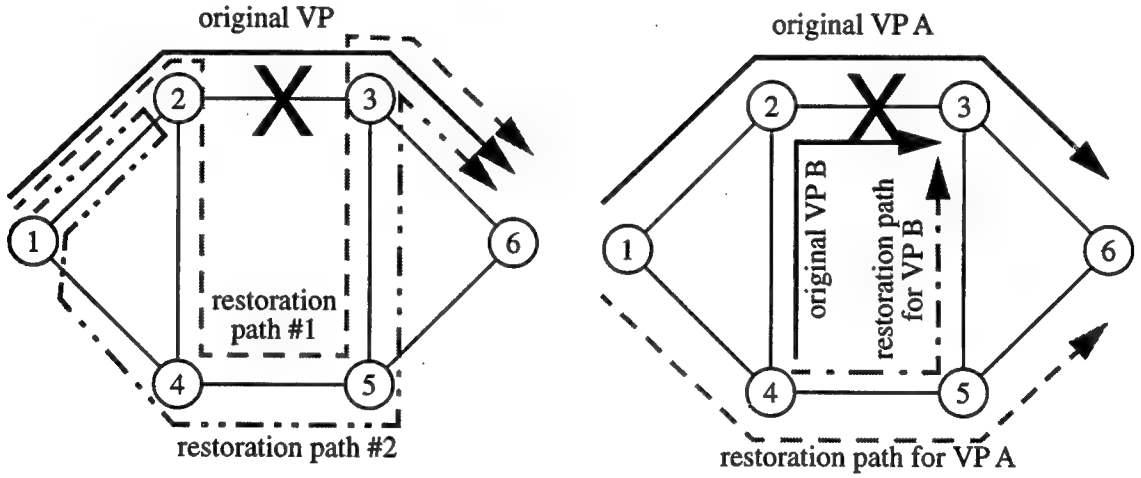
(a) *k*-shortest path-based restoration (b) maximum flow-based restoration

Figure 2-4. Two restoration path selection strategies

Assume all working links possess a unit spare bandwidth. Then only a unit flow is restored with *k*-shortest path-based restoration, while two unit flows are recovered with maximum flow-based restoration (restoration paths are shown in bold lines). This example is taken from [20]. This type of network configuration is named a 'trap' topology in the reference. They attributed to it the failure for the *k*-shortest path strategy to find a maximum flow assignment.

With restoration path preplanning, the MF restoration scheme could realize even faster recovery than the KSP restoration scheme. However, a larger memory space is required at each node and, more importantly, it is difficult to keep up-to-date restoration information in a frequently changing network environment. On the other hand, the KSP restoration protocol does not require any knowledge of the current spare bandwidth distribution over the network. Therefore, it would be preferable for networks involving a frequent change in the VP-level demand patterns.

Depending on the location where traffic rerouting is performed, the network restoration strategies can be further categorized into two classes: line restoration and end-to-end restoration (Figure 2-5). When a link failure occurs, a line restoration scheme dispatches alternate routes between the two end-nodes of the failed link and reroutes all affected traffic around the link. On the other hand, the end-to-end restoration scheme switches failed VP's to alternate routes established between their respective source and destination nodes. When a failure is detected, recovery messages are sent to the source and destination nodes of all affected VP's to inform the failure. At this time, the bandwidth occupied by the affected VP's is released over their original routes. Then, each source



(a) Line restoration

(b) End-to-end restoration

Figure 2-5. Line restoration and end-to-end restoration

Upon a failure of link 2-3, a VP 1-2-3-6 is restored via 1-4-5-6 with end-to-end restoration. However, line restoration would restore the same VP over a longer path 1-2-4-5-3-6 (restoration path #1) or over a considerably more inefficient path 1-2-1-4-5-3-6 which includes backhauling [5] on link 1-2 (restoration path #2). All affected VP's are rerouted between Nodes 2 and 3 with line restoration. On the other hand, rerouting usually takes place at different locations with end-to-end restoration. In this example, the VP A is rerouted between Nodes 1 and 6, while the VP B is recovered between Nodes 4 and 3.

node initiates virtual path switching over alternate routes. Note that all affected VP's are rerouted between the same end nodes with line restoration (between Nodes 2 and 3 in the example of Figure 2-5-a), while path rerouting takes place at different locations with end-to-end restoration (see Figure 2-5-b).

End-to-end restoration generally requires a lower resource installation cost to realize a fully restorable network since spare bandwidth can be utilized more efficiently [5] [43] [82]. For given network resources, it has the potential of attaining a higher restorability from a failure. Furthermore, end-to-end restoration can be applied to recovery from a node failure as well as to that from a link failure [43]. On the other hand, line restoration is expected to be faster in terms of restoration time because its restoration paths are generally shorter and involve fewer intermediate nodes. However, the VP activation process (Phase 4) could be a bottleneck for line restoration due to the localization of the process around a failed link. End-to-end restoration, on the contrary, could be achieved fast enough since the rerouting task can be distributed widely over the network [5]. This argument is based on the assumption that the restoration paths are precalculated. The computational complexity of the rerouting path decision process is substantially higher for the end-to-end restoration scheme[2].

According to the above classification, there are four possible variations for fast restoration protocols: (1) MF-based rerouting with line restoration (MF-LINE), (2) KSP-based rerouting with line restoration (KSP-LINE), (3) MF-based rerouting with end-to-end restoration (MF-ETE), and (4) KSP-based rerouting with end-to-end restoration (KSP-ETE). Among these combinations, however, the KSP-ETE rerouting appears to be impractical due to a very slow restoration speed. Since path rerouting takes place between source and destination nodes in the KSP-ETE scheme, the restoration messages must traverse in a considerably wider area compared to the KSP-LINE scheme. Thus, a higher hop count should be specified to reach the destination, but this increases the number of broadcast messages. Moreover, this message flooding procedure must be performed individually for all source and destination node pairs of the virtual paths affected by a failure. As a result, the number of recovery messages could explode over the network, which would significantly delay the restoration procedure. For this reason, the KSP-ETE is not considered in our study[3].

2. Given a spare bandwidth assignment, line restoration seeks restoration paths between two end nodes, while end-to-end restoration attempts to find restoration paths between multiple node pairs. Thus, the former problem is formulated as a single commodity flow problem. On the other hand, the rerouting path hunting problem for end-to-end restoration is a multicommodity flow problem, whose complexity is significantly higher than that of a single commodity flow problem.

2.1.3. ATM resource management

In ATM networks, resource allocation requests arise at various levels: ATM cells request a time slot of a transmission link, calls contend for a virtual path bandwidth, and virtual paths claim physical link capacity. An ATM resource management system must handle these requests so that quality of service (QOS) objectives are satisfied. For example, cell loss rate and call blocking probability must be enforced to be less than their designated levels, and virtual path reconfiguration requests must be granted at a higher possibility than a specified level. A physical resource must also be properly designed in order to meet these goals. Finding the best resource assignment strategy, however, is a very complicated task since several different levels of requests must be controlled under various types of QOS objectives.

In order to reduce the complexity, a layered resource management architecture has been proposed for the ATM networks [40]. In their architecture, resource allocation is layered by time scales. Arrivals of cells, calls, VP reconfiguration requests and physical resource allocation requests occur on different time scales. The processing time of a cell is on the order of micro-seconds or less, a call session lasts for minutes, a virtual path configuration changes every few hours or days, and a new physical resource allocation is designed for once a year or over a longer interval. Typically, their time scales differ by two or more orders of magnitude. Based on this observation, the resource allocation control is layered in four layers: a cell layer, a call layer, a virtual path (VP) layer and a facility network (FN) layer (refer to Figure 2-6).

Resource assignment at a layer is performed so that the QOS of the next lower layer is guaranteed. When a new call arrives, a call layer reserves sufficient bandwidth to assure the cell level QOS (i.e, cell loss rate objective) for the connection. A virtual path layer performs reconfiguration so that enough virtual path bandwidth is allocated to satisfy the call level QOS (i.e. call blocking rate objective). Finally, a facility network layer designs physical resource placement so that the probability of failure to meet virtual path bandwidth request (i.e. VP-level QOS) is small.

The heterogeneity of traffic supported in the ATM network further complicates the traffic con-

3. Lin et. al. [52] discuss in their recent paper a new restoration protocol which is a variant of KSP-ETE. It calculates candidate alternate VP's beforehand. When a failure happens, the available bandwidth is searched only over the preassigned routes. Their simulation result indicates that restoration can be completed mostly in a second if the protocol is applied to a metropolitan LATA network. In other words, it might be difficult to adapt it to larger networks even with this precaution.

trol. The network must support a wide range of traffic with different bandwidth requirements and different degrees of burstiness. In a connection of variable bit rate service (e.g. compressed video traffic), the source bit rate varies over time, and this creates bursty cell arrivals. The notion of *equivalent bandwidth* has been proposed to facilitate the bandwidth management of variable bit rate sources [21] [38] [40]. The equivalent bandwidth provides a unified metric representing the effective load for the connection. It is computed based on the source statistics and requested QOS of the connection.

The proposed architecture considerably simplifies the complexity of the ATM network management. Due to a large difference in the operating time scales between two adjacent layers, the transmission resource configuration at a higher layer can be regarded to be quasi-static. Thus, each layer only has to manage the corresponding traffic entity[4] for a given view of an available higher-level network resource [40]. For example, a call admission and its routing can be determined at the call layer for a given VP sub-network, and a virtual path configuration can be optimized at the VP layer for a given physical network. With this limited view of a network resource, the network manager at each layer can concentrate on resource allocation of its traffic entity to promote the QOS of its own layer. For example, the call layer can employ dynamic call routing to reduce call blocking in a VP sub-network [7] [46].

Another simplification attained by the layered architecture is in the interaction between layers. Each layer only has to care about assuring the QOS objective of the next lower layer. This QOS assurance is accomplished by assigning sufficient bandwidth at the layer. The equivalent bandwidth can be defined at each layer, and it expresses the required bandwidth to guarantee the QOS up to the layer of concern [40]. For example, consider the equivalent bandwidth defined at the VP layer, which we call a *VP-level bandwidth* or *VP-level traffic demand*. By satisfying VP-level traffic demand, a resulting VP sub-network can restrict the call blocking rate below its designated level. In the formulation of the survivable virtual path routing (SVPR) problem as well as the survivable capacity and flow assignment (SCFA) problem discussed in the following sections, the end-to-end flow requirement is assumed to be given by a VP-level bandwidth. The aggregate traffic load can be obtained by simply adding the equivalent bandwidth of each connection [21]. Using this property, we can define the problem as a multicommodity flow problem [9].

---

4. Traffic entity can be cells, calls or virtual paths.

2.2. Survivable ATM Network Management

2.2.1. Survivable ATM network management architecture

Network survivability becomes increasingly important as society becomes more dependent on communication networks. However, most previous work on ATM traffic control dwells on an implicit assumption that all network components are functioning. In order to build a reliable ATM network, survivability functions must be incorporated into an ATM resource management system. Their integration, on the contrary, increases the complexity of the ATM network management tasks. To overcome the difficulties, the layered management architecture described in the previous section is extended to survivable ATM network management. The architecture can simplify the resource management by classifying various levels of network resources and traffic entities into layers.

Figure 2-6 illustrates the survivable ATM network management architecture proposed in this project. The survivability functions are embedded at the VP and higher layers, considering the fact

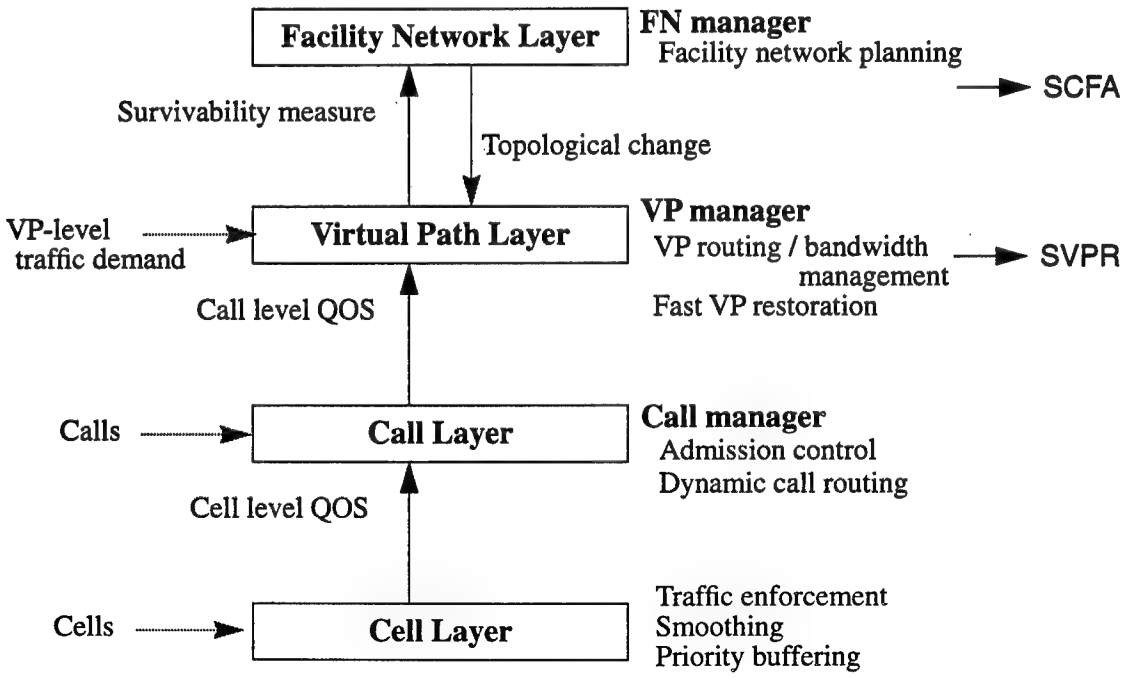


Figure 2-6. A survivable ATM network management architecture

that path level recovery enables rapid and efficient restoration and reduces the complexity of traffic management [82] (refer to Section 2.1.2). A *survivability measure* (see Section 2.2.3 for details) is introduced to capture the degree of attainable network survivability. At the VP layer, the *VP manager* builds a VP sub-network over currently available physical network resources. Given a VP-level traffic demand satisfying call level QOS, the VP manager configures virtual paths so that the survivability measure is optimally enhanced. This leads to the SVPR problem discussed in Section IV. The VP manager also performs fast VP restoration when a network failure happens. If the VP manager cannot maintain a survivability measure at a desired level due to a growth of traffic demand, the FN layer must trigger a facility network planning process. The *FN manager* designs an additional physical resource placement for a newly projected traffic demand so as to satisfy the survivability measure objective[5]. The SCFA problem discussed in the next section deals with this physical network design problem.

2.2.2. Two-step restoration scheme

Two contradicting requirements should be satisfied upon virtual path restoration. After a failure, it is desirable to realize an optimal VP configuration which incurs the least service interruption upon a possible subsequent failure. However, the optimal flow calculation introduces a computational delay, while it is essential to complete the restoration as quickly as possible for high-speed networks. In order to accommodate these contradicting requirements, fast VP restoration and optimal VP reconfiguration, the VP manager uses a *two-step restoration* approach as shown in Figure 2-7. Upon failure, the *VP restoration manager* executes the fast restoration procedure to accelerate a recovery from the failure. After the restoration is completed, the *VP planner* module computes an optimal VP assignment for the new network topology, and a VP configuration is again changed to the newly calculated optimal solution (*network-wide restoration*). Although this scheme temporarily produces a flow which is not optimal from the survivability viewpoint, it is permissible in practice since more than one network failure may not happen successively in a short time.

5. The VP-level QOS objective must also be satisfied by the FN layer. Namely, the probability of failure to assign requested VP-level bandwidth must be less than a designated level. Since a survivable network requires significantly more physical resource installation for restoration purpose, however, a network satisfying the survivability measure objective can also meet the VP-level QOS objective. Therefore, it is not necessary to consider the VP-level QOS in the survivable network management system.

2.2.3. Survivability measure

The survivability measure gives a quantitative metric for the attained survivability level of the network. It is used as a decision criterion for resource management control in a survivable ATM network. For example, the VP manager employs the survivability measure as an objective function of the VP planner module. Based on this measure, the FN manager decides when to initiate a facility network planning process. The survivability measure is also useful in evaluating the performance of the proposed VP planner modules.

A good measure of survivability should express the actual amount of damage experienced in the network or by end-users, instead of using traditional reliability criteria such as global availability. A number of lost calls have been proposed for such a measure at the call layer in telecommunication networks [55] [70]. Similarly, the amount of lost flow (VP-level bandwidth) can be employed to express damage to the VP layer. A restoration ratio has also been used in the literature of self-healing networks [37] [78]: It is defined as an expected percentage of recoverable flow per span failure. Since the average flow per span depends on flow assignment, this measure does not precisely express the amount of damage in the dynamically reconfigurable network environment. In the following, the amount of lost flow is employed as the survivability measure.

Let S be a set of possible failure patterns and w_s be the probability of a failure event $s \in S$.

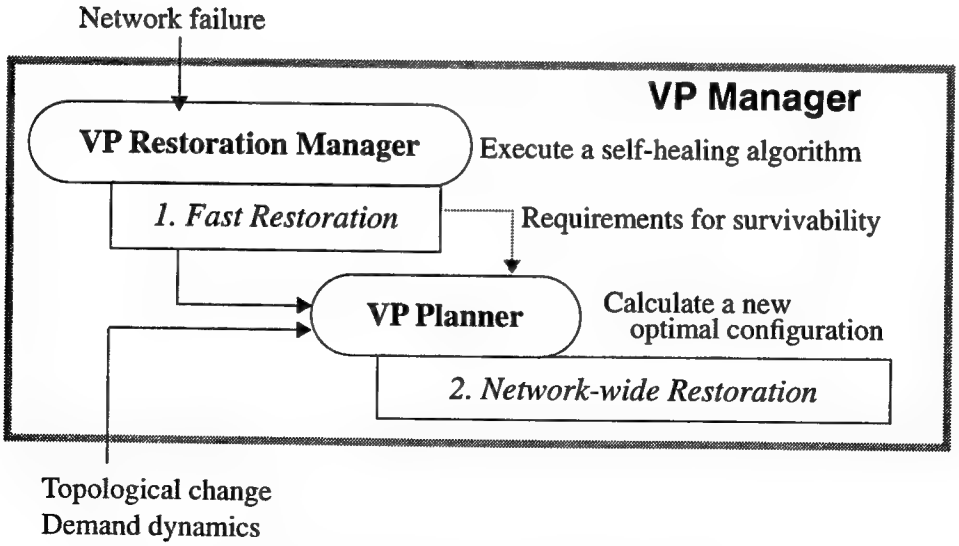


Figure 2-7. Two-step restoration approach

Then, the *expected lost flow L*, given by $L = \sum_{s \in S} w_s \times L_s$, can be employed as the survivability measure, where L_s is a lost flow due to a failure event $s \in S$. The selection of the state space S is another factor for survivable network design. The more failure patterns are included, the more precise estimate of the damage the survivability measure can express. However, this increases the complexity of the design procedure. For design purposes, it would be sufficient to consider a set of the most probable failure events as a whole state space. According to [82], a complete fiber cable cut is the most common and frequently reported failure event among disastrous network failures over the decades. Therefore, we take the events of a single link failure as a whole state space (S=E) and assume that each failure is equally weighted ($w_s = 1/|E|$), where E is a set of links in the network. Note that although a VP assignment is computed assuming a single link failure, the system can work against any failure as long as the fast restoration algorithm supports it.

A worst lost flow per failure, given by $L = \max_{s \in S} \{L_s\}$, would be another candidate for representing the survivability measure. By minimizing this measure, the VP planner module will find an optimal minimax routing. The minimax routing has been reported to be preferable since the optimal solution tends to balance the traffic over the network, resulting in better call and cell level QOS [48]. However, it turns out to be inappropriate for the survivability measure. The problem arises if there is a link where a significant loss becomes inevitable due to a demand growth. A typical example is a failure of either link adjacent to a node of degree two. If line restoration is employed, all affected VP's must be rerouted over the other remaining link incident to the node. Thus, if the total bandwidth of either link is less than the demand originating and terminating at this node, then data loss is unavoidable after a failure of the adjacent link[6]. Maximum loss per failure would occur upon failure of such a link. Then, any flow assignment can be the optimal solution based on the minimax function, as long as no virtual paths are relayed at the node to minimize this maximum loss. However, the attainable survivability level with other link failure scenarios could significantly differ among such assignments, but the minimax objective function cannot distinguish them.

2.2.4. Network reconfiguration

The primary goal of network reconfiguration is to realize a high utilization of network resources while maintaining the design objectives of each layer (QOS or survivability measure) at an accept-

6. Namely, the only remaining link leaving the node cannot accommodate the demand originating and terminating at the node.

able level. Three types of network reconfiguration are identified corresponding to the FN, VP and call layers. Network reconfiguration mechanisms at the FN and VP layers are related to the survivability requirements, which lead to the SCFA and SVPR problems discussed in the subsequent sections.

Each call session typically lasts for minutes. Thus, an aggregate traffic demand may fluctuate over minutes to hours. This short-term demand fluctuation can be accommodated at the call layer by applying a dynamic call routing mechanism over a VP subnetwork. Dynamic call routing attempts to use an alternate route when a primary route (e.g. direct VP) is not available (refer to [7] [8] and [46] for details). We refer to this dynamic adjustment of call routing as *short-term reconfiguration*. By adaptively assigning transmission resources, dynamic call routing is expected to cope with load variations and to lower a call blocking rate. Furthermore, the peak load at a different time-zone occurs at a different time in a long-haul network. A dynamic routing scheme routes a part of the peak traffic at one-zone over alternate routes in a non-peak zone in order to promote resource utilization [46]. The VP-level reconfiguration is required when the call level QOS cannot be maintained even with the dynamic call routing mechanism due to a further traffic fluctuation over a longer time scale.

VP-level dynamic reconfiguration, which is called *long-term reconfiguration*, promotes the attainable survivability level in response to a change in network environments. In addition to the network-wide restoration after a failure, there are several scenarios in which the VP planner is invoked. First of all, the VP planner is triggered when the higher layer configuration (facility network) has changed, not only due to a network failure (an unplanned change) but also due to a planned change such as reintegration of the repaired component, a new facility installation, or a planned facility removal (for a component test, for example). VP reconfiguration is executed in order to cope with a change in the underlying facility network and to improve the network survivability. VP reconfiguration process is invoked before the planned removal of a network resource, or after the planned installation or unplanned change. These changes would happen very infrequently, say over months.

A dynamic VP reconfiguration procedure is invoked more frequently to cope with demand dynamics. The VP-level traffic demand patterns may change in hours due to demand variations within a day. It may also fluctuate in days or weeks due to the daily or seasonal demand character-

istics or due to a long-term growth of traffic demand. Owing to a vast volume of traffic aggregated in the ATM inter-office networks and due to the effect of dynamic call routing, it is expected that short-term load variations would be statistically averaged. Thus, the VP reconfiguration interval would be relatively long: It could be a few hours or days depending on how fast the aggregate traffic fluctuates [40].

The reconfiguration interval is also influenced by the VP traffic control policy and the speed of the VP planner module. A fine tuning of the VP configuration could lead to a higher utilization of the network resource. However, a frequent update increases the load of the VP planner module. Thus, the update interval must be determined considering the trade-off between reducing processing complexity and increasing network utilization. The VP planner should compute a new VP configuration as quickly as possible, say in a period one or two orders of magnitude less than the VP update interval. Thus, the speed of a VP planner module further determines a feasible range of a VP configuration update interval.

The *call manager* at the call layer monitors call-level traffic activities and periodically calculates statistics on a call blocking rate as well as VP bandwidth usage. VP reconfiguration is required if a call blocking rate exceeds the call level QOS objective. The VP reconfiguration process is also necessary if the demand forecast for an upcoming period[7] cannot be accommodated in the current configuration. In these cases, VP planner reconfigures virtual paths so as to satisfy a newly projected VP-level traffic requirement. Out of such candidate configurations, the VP planner seeks the one which attains the best network survivability.

When a new VP configuration is calculated, the change must be made gracefully. Here 'graceful' means that the VP reconfiguration would not affect the connections currently in service. In order to avoid abrupt service interruption and provide service transparency, the cell sequence integrity must be assured for all VC's upon VP reconfiguration. If a VP route stays the same, this precaution is unnecessary; VP reconfiguration can be accomplished immediately by changing its bandwidth. However, the VP reconfiguration may involve a radical change on its routing assignment, especially upon the network-wide restoration after a failure. In such a case, cells may arrive out of sequence if a path is switched over without any precaution. There are two possible

7. The demand projection could be calculated from the current traffic load and the demand patterns experienced in the past. The maximum required bandwidth over an upcoming period is reserved.

approaches to realize graceful VP reconfiguration: One way is to keep an old virtual path until all active connections are closed. All newly admitted calls are routed over a new path, and the VP bandwidth is gradually reassigned to the new path as the VP bandwidth over the old path is released. Another possible approach is the application of the ATM hitless path switching mechanism [76] [82]. The hitless switching mechanism synchronizes between the old and new routes. The difference of delays between the two routes is adjusted by cell buffering if the original route is longer than the new one[8]. This hitless switching mechanism can be gradually applied to virtual paths involving their route change. Combination of the above two approaches would give a further alternative, where long-lasting connections over an old virtual path are finally switched to a new path by the hitless path switching mechanism.

Finally, the network reconfiguration also occurs at the FN layer. When the VP planner cannot maintain survivability measure at a desired level[9], the FN manager is invoked to design a survivable network in a cost-effective manner. This network reconfiguration is called *very long-term reconfiguration*. Given a maximum VP-level traffic demand projected over some future time frame (say a year or two), the FN manager seeks the physical resource assignment which guarantees full restorability from network failure at a minimum cost. This reconfiguration typically happens on a very long time scale, say over years.

Table 2-1 summarizes various types of the network reconfiguration for survivable ATM networks.

2.3. Summary

This section introduces the proposed survivable ATM network management system. A survivable ATM network management architecture integrates survivability functions into the existing ATM resource management system at the FN and VP layers. In order to achieve a higher level of network survivability, network reconfiguration is performed at these two layers. This leads to the two problems discussed in the following two sections: the SCFA (survivable capacity and flow assignment) problem and the SVPR (survivable virtual path assignment) problem. The survivabil-

8. In this case, out-of-sequence cell delivery may happen without buffering since a cell over a new route just after path switching may arrive at the destination earlier than a cell over an old route just before the switching.

9. For example, when full restorability is not assured for a long period (say a few months), or when an expected lost flow (survivability measure) averaged over months grows beyond a designated level (say 0.1 percent of the average load). The statistics on the attained survivability measure can be collected by the VP manager.

ity measure is introduced to control resource allocation in the proposed survivable network management system. The two-step restoration strategy is proposed to meet two contradicting requirements of virtual path restoration: fast restoration and optimal flow assignment. Its effectiveness will be discussed in Section IV.

This section presents several implementation techniques of the fast restoration protocol. The selection of a restoration strategy completely depends on the application environment. The following criteria must be considered at the design phase of a survivable ATM network:

1. Link installation cost
2. Attainable restorability for a given physical resource
3. Speed of the VP planner module
4. Speed of the VP restoration manager module
5. Frequency of change in the VP-level traffic demand
6. Implementation cost of the VP manager

This project clarifies the effectiveness of each restoration scheme with respect to the first three criteria. Required spare capacity cost is elaborated in the next section, while Section IV discusses the attainable restorability of each scheme as well as the applicability of the proposed VP planner module based on the required computation time. The analysis presented in this project would help to elucidate the applicable VP control and restoration strategies. The results can be applied to the decision process at the design phase of a survivable ATM network management system.

Table 2-1. Various types of network reconfiguration for survivable ATM networks

Type of network reconfiguration	Descriptions
Very long-term reconfiguration	Reconfiguration of the underlying facility network at the FN layer to maintain the survivability level. Triggered when a VP planner cannot maintain a survivability measure at a satisfactory level, mainly due to a growth of traffic demand over a very long time scale, say over years.
Long-term reconfiguration	Network-wide optimal VP reconfiguration. Triggered mainly due to a change in a traffic trend at a longer time scale, say over hours or days. Also invoked when a significant topological change occurs, including one due to a failure.
Short-term reconfiguration	Reconfiguration due to traffic fluctuation for minutes to hours. Use a dynamic call routing mechanism over a VP subnetwork at the call layer.

III Survivable Capacity and Flow Assignment (SCFA)

The SCFA problem arises when the virtual path layer cannot maintain the network survivability at a desired level due to a growth of traffic demand over months or years. The facility network layer invokes the FM manager to plan additional network resources on such occasions. This section develops optimization procedures for the networks based on two different restoration schemes: line restoration and end-to-end restoration. Capacity and flow assignment is jointly optimized to find an optimal capacity placement. Several mechanisms are proposed to reduce the computational complexity of the joint optimization. As discussed in Section II, end-to-end restoration schemes have been considered more advantageous than line restoration schemes because a fully restorable network could be realized with less spare capacity. A comparative analysis is extensively conducted to clarify the benefit of end-to-end restoration schemes quantitatively in terms of minimum resource installation cost. This study reveals that the economical gain could be nominal for a well-connected and/or unbalanced network.

3.0. Introduction

The SCFA problem aims to find the most economical capacity placement which assures a fully-restorable virtual path assignment for a projected traffic demand. Previously, only a limited number of works have studied an optimal spare capacity placement problem for line restoration-based self-healing STM networks [37] [68]. Given the number of working transport paths per link, these studies calculate a spare link assignment with minimum installation cost, subject to the constraint that any single-link failure can be restored successfully. A flow is preassigned to each link, although there is no guarantee that this assignment gives the most cost-effective capacity placement. As for end-to-end restoration-based networks, no published work on the optimal capacity assignment problem was found.

We propose a jointly optimal capacity and flow assignment procedure in order to find a globally optimal solution for a given link cost function. Since there is mutual dependency between the link capacity placement and the survivable flow assignment with a given link capacity, the obtained

solution cannot be claimed to be a true optimum if these problems are treated separately [24]. However, the complexity of the problem grows tremendously with joint optimization. Several mechanisms are developed to make this problem computationally more tractable.

Despite the high complexity of its rerouting decision process, an end-to-end restoration scheme has been widely cited to be more advantageous than a line restoration scheme [5] [11] [19] [43] [68] [82]. The former scheme could effectively use the spare bandwidth, and thus less redundant capacity would be necessary to construct a fully restorable network. However, no comprehensive work has been performed on the quantitative comparison of these two schemes to confirm this assertion for various network environments. The only available numerical results so far are based on computer simulation[1] [5] or heuristics [43] using a single sample network. However, it is not clear how close the obtained solution is to the optimum. The comparison based on heuristics leaves us uncertain as to whether the benefit comes from the end-to-end restoration scheme or the optimization error.

With globally optimal solution procedures at hand, we can now make a meaningful comparison between the two restoration schemes. A comparative study is performed with respect to the minimum spare capacity installation cost in order to elucidate the economical gain due to the end-to-end restoration schemes. Several networks with diverse topological characteristics as well as several projected traffic demand patterns are employed in the experiments to see the effect of various network parameters.

The rest of the section is organized as follows: Section 3.1 provides the formal definition of the SCFA problem and discusses several options which determine the problem structure. The problem formulation and the solution approach are presented in Section 3.2 for the self-healing networks based on line restoration protocol, while Section 3.3 is devoted to those based on end-to-end restoration protocol. The validity of the proposed algorithms is also demonstrated in the respective sections. The results of comprehensive numerical study are reported in Section 3.4, followed by the concluding remarks in Section 3.5.

1. No details are provided in the reference.

3.1. Formal Problem Definition

An ATM network is modeled as a directed graph $G = (V, A, \underset{\sim}{c})$ where V is a node set representing ATM switches, A is a set of directed arcs representing optical trunks, and $\underset{\sim}{c} = (c_a)$ is a vector of arc capacity ($a \in A$). Let E denote a set of undirected links. We assume that the network is bidirectional and that each link consists of two directed arcs with the same end-nodes but in opposite directions (i.e. $|A| = 2 \cdot |E|$). A link represents a complete set of optical fibers installed between two nodes, and a network failure (e.g. a complete span cut) is expressed in terms of a link in the problem formulation. On the other hand, an arc embodies a set of transmission media going from one node to the other end of a link, and the capacity and flow assignments are calculated per arc.

A commodity is a traffic flow from an origin to a destination. One commodity is defined for each origin and destination pair. Let Π be a set of commodities in the network and $Q = (q^{\pi})$ be a vector of the requested VP-level bandwidth for each commodity $\pi \in \Pi$. Multiple VP's are established for each commodity to satisfy the traffic requirement. It is assumed that each commodity involves a significant number of VP's and thus that the bandwidth of each VP is considerably small compared to q^{π}. This is a reasonably realistic assumption for the future ATM interoffice exchange networks where a very large volume of calls with a wide variety of services are aggregated. A VP is expected to have a bandwidth on the order of a few Mbps to a hundred Mbps [6], while a link will be composed of multiple optical fiber cables, each having a capacity on the order of ten Gbps to Tbps [56] [66]. Based on the above assumption, a commodity flow as well as a restoration flow can be simply expressed by a single continuous flow variable in the problem formulation, instead of using an individual variable per VP. This considerably reduces the complexity of the problem while maintaining the accuracy of the solution. Let $\underset{\sim}{f}$ denote a vector of arc flow.

Now the SCFA problem can be formally stated as the following optimization problem:

Given	V, A and Q,		**(SCFA)**
Minimize	Network resource installation cost $D(\underset{\sim}{c})$		
over	$\underset{\sim}{c}$ and $\underset{\sim}{f}$,		
subject to	a) $\underset{\sim}{f}$ is a multicommodity flow satisfying Q,	(flow conservation law)	

b) capacity constraints are satisfied, and (capacity constraints)

c) full restorability is assured through fast restoration.

(full restorability constraints)

The full restorability constraints differ according to the failure events considered in the network design as well as the traffic rerouting strategy adopted in the self-healing network. From the survivability viewpoint, a network should be designed to fully recover from as many failure patterns as possible. However, this will increase not only the network installation cost but also the complexity of the optimization procedure. As discussed in Section 2.2.3, a complete span cut is considered in our formulation since it is known to be the most common and frequently reported failure event over the decades [82].

The full restorability constraints further differ according to the traffic rerouting strategies. Three strategies identified in Section 2.1.2 are considered in the SCFA problem: (1) MF rerouting with line restoration, (2) MF rerouting with end-to-end restoration and (3) KSP rerouting with line restoration. In the following, the corresponding SCFA problems are referred to the SCFA-MF-LINE, SCFA-MF-ETE and SCFA-KSP-LINE problems, respectively. As will be discussed in Section 4.4, the KSP-based formulation suffers from non-convex constraints. The MF-based formulation, on the other hand, is immune to this problem, and the constraints can be written as a set of linear equations by introducing restoration flow variables. In the following, we focus on the MF-based systems, and their globally optimal solutions are obtained through the mathematical programming techniques. We develop a heuristic approach for the SCFA-KSP-LINE problem. The initial solution of the SCFA-KSP-LINE is obtained from the optimal solution of the SCFA-MF-LINE problem.

3.2. SCFA-MF-LINE

3.2.1. Problem formulation

The intricate math is shown in Appendix A in order to preserve the flow of the document.

3.2.2. Solution approach

The intricate math is shown in Appendix B in order to preserve the flow of the document.

3.2.3. Validity of the algorithm

The intricate math is shown in Appendix C in order to preserve the flow of the document.

3.2.4. Extension to SCFA-KSP-LINE

The intricate math is shown in Appendix D in order to preserve the flow of the document.

3.3. SCFA-MF-ETE

3.3.1. Problem formulation

The intricate math is shown in Appendix E in order to preserve the flow of the document.

3.3.2. Solution approach

The intricate math is shown in Appendix F in order to preserve the flow of the document.

3.3.3. Validity of the algorithm

The intricate math is shown in Appendix G in order to preserve the flow of the document.

3.4. Evaluations

3.4.1. Major advantages of the ETE and JOA schemes

The required resource installation cost is expected to decrease with joint optimization as well as end-to-end restoration. In order to understand the causes of the economical benefit due to these schemes, we first analyze the results based on the small sample network shown in Figure 3-1. The two optimal assignment approaches are employed for each restoration scheme: Jointly Optimal Assignment (JOA) and Optimal Capacity Assignment (OCA). The JOA jointly optimizes the capacity and flow assignment using the techniques proposed in the preceding two sections, while the OCA optimizes the capacity placement with a predetermined commodity flow assignment. The commodity flow is fixed to be the shortest route flow[2] in our experiment. Since the shortest route flow requires the least amount of capacity to satisfy the traffic demand, this flow assignment must

2. A shortest route flow is a multicommodity flow where each commodity is routed over its shortest path under a certain arc cost [22]. In this case, an arc cost is defined to be a unit capacity cost over an arc.

be the best choice if a flow must be predetermined[3]. Thus, a comparative study between the JOA and OCA schemes clarifies the benefit of the joint optimization. The capacity and flow assignment of the following four cases are, therefore, studied in this experiment: 1) end-to-end restoration with JOA (ETE-JOA), 2) end-to-end restoration with OCA (ETE-OCA), 3) line restoration with JOA (LINE-JOA), and 4) line restoration with OCA (LINE-OCA). The traffic demand used in the experiment as well as the resulting commodity flow assignment are shown in Table 3-2. Table 3-3 summarizes the optimal capacity placement for each scheme. A restoration flow assignment is illustrated for two failure scenarios in Figures 3-2 and 3-3. Three major causes are identified for the superiority of the end-to-end restoration scheme.

Cause 1. With the line restoration scheme, a VP may suffer backhauling [5] after restoration. Moreover, it could even contain a loop. On the other hand, end-to-end restoration is immune to these problems, and thus it can avoid unnecessary consumption of spare bandwidth.

Consider the case of the OCA approach where the flow assignments for the two restoration schemes are identical. Upon a failure of link 2, the <5,1> commodity[4] is restored over the path 5-3-2-1[5] with end-to-end restoration (Figure 3-2-b). Line restoration recovers the <5,1> commodity over 5-4-5-3-2-1. Thus a flow is assigned to traverse back and forth between nodes 5 and 4 (Figure 3-2-a). This phenomenon is called backhauling [5], and the spare bandwidth over 5-4-

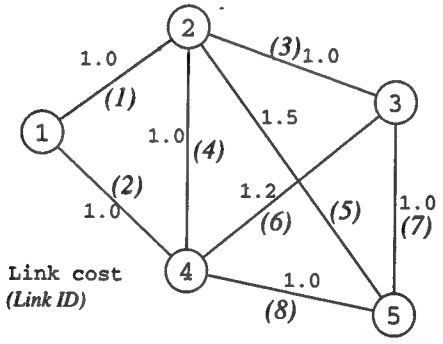


Figure 3-1. A small sample network

3. This hypothesis is further confirmed by the following result with the JOA-based assignment: At the optimum, virtually all commodities are served via a single route to carry their traffic demand, and only a few use multiple routes. Although such a route is not necessarily the shortest one, most flows are routed over the least expensive path. This suggests that the shortest path is the best choice for most commodities, and thus a shortest route flow is an obvious choice if a flow must be predetermined.

4. A commodity from a source node *a* to a destination node *b* is referred to as commodity <*a*,*b*>.

5. *a*-*b*-*c* denotes a path from nodes *a* to *c* through *b*.

Table 3-1. Demand and commodity flow assignment in the small sample network

Src.	Dest.	Demand	ETE JOA	LINE JOA	OCA	Route
1	2	200	200	200	200	1 -> 2
2	1	200	200	200	200	2 -> 1
1	3	100	100	100	100	1 -> 2 -> 3
3	1	100	100	100	100	3 -> 2 -> 1
1	4	300	300	300	300	1 -> 4
4	1	300	300	300	300	4 -> 1
1	5	100			100	1 -> 4 -> 5
			100	100		1 -> 2 -> 5
5	1	100	100	100	100	5 -> 4 -> 1
2	3	200	200	200	200	2 -> 3
3	2	200	200	200	200	3 -> 2
2	4	1,000	700	950	1,000	2 -> 4
			300			2 -> 1 -> 4
				50		2 -> 3 -> 4
4	2	500	500	500	500	4 -> 2
2	5	200	200	200	200	2 -> 5
5	2	200	200	200	200	5 -> 2
3	4	300	300	300	300	3 -> 4
4	3	300	200	300	300	4 -> 3
			100			4 -> 5 -> 3
3	5	100	100	100	100	3 -> 5
5	3	100	100	100	100	5 -> 3
4	5	300	300	300	300	4 -> 5
5	4	300	300	300	300	5 -> 4

The demand and commodity flow assignment is given in terms of BU (bandwidth unit). The BU is an arbitrary unit of bandwidth, say Mbps.

Table 3-2. Optimal arc capacity assignment in the small sample network

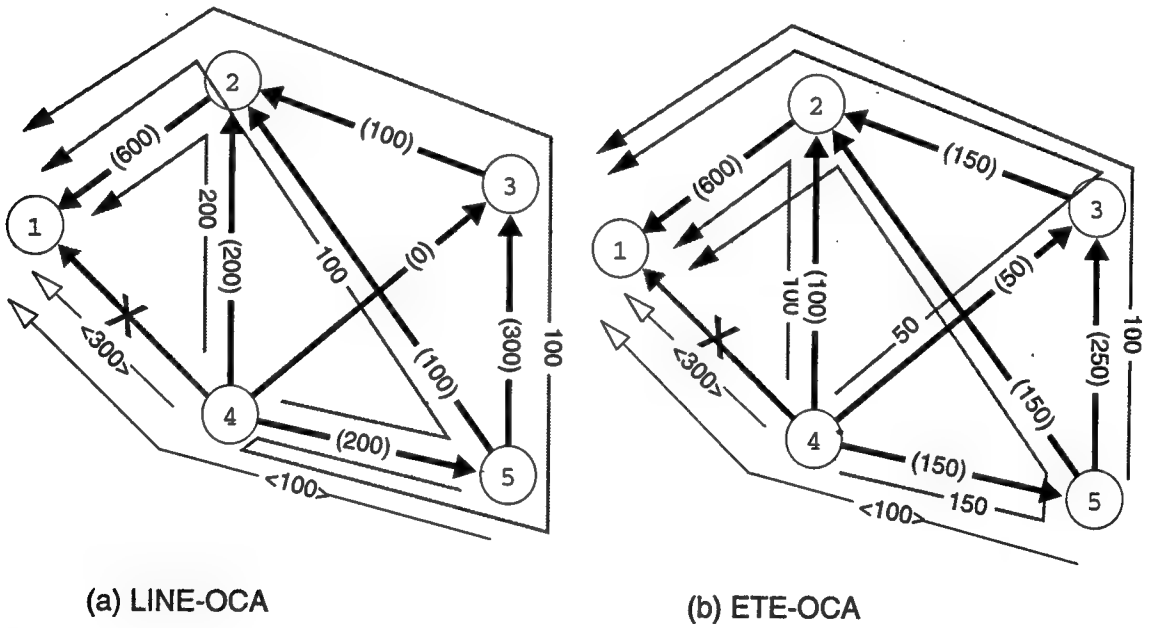
Arc (src. - sink)		1-2	2-1	1-4	4-1	2-3	3-2	2-4	4-2	2-5	5-2	3-4	4-3	3-5	5-3	4-5	5-4	Total
Arc cost[b]		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.5	1.5	1.2	1.2	1.0	1.0	1.0	1.0	Cost[a]
ETE JOA	Total[c]	700	700	700	700	700	500	700	700	500	200	600	200	200	500	600	700	9410
	Flow[d]	400	600	600	400	300	300	700	500	300	200	300	200	100	200	400	400	6250
	Spare[e]	300	100	100	300	400	200		200	200		300		100	300	200	300	3160
LINE JOA	Total	700	900	900	700	550	500	950	700	450	200	550	300	250	500	500	550	9695
	Flow	400	300	300	400	350	300	950	500	300	200	350	300	100	100	300	400	5930
	Spare	300	600	600	300	200	200		200	150		200		150	400	200	150	3765
ETE OCA	Total	700	900	1000	700	450	450	1000	600	450	350	450	350	250	350	550	650	9760
	Flow	300	300	400	400	300	300	1000	500	200	200	300	300	100	100	400	400	5820
	Spare	400	600	600	300	150	150		100	250	150	150	50	150	250	150	250	3940
LINE OCA	Total	700	900	1000	700	500	400	1000	700	400	300	500	300	300	400	600	600	9810
	Flow	300	300	400	400	300	300	1000	500	200	200	300	300	100	100	400	400	5820
	Spare	400	600	600	300	200	100		200	200	100	200		200	300	200	200	3990

a. Total installation cost (CU). CU (cost unit) is an arbitrary unit of arc capacity installation cost.
b. A unit arc cost (CU/BU).
c. Total capacity of an arc (BU).
d. Aggregate arc flow. (BU)
e. Spare capacity of an arc. (BU)

5 is wasted. In our experiment with a larger network, line restoration could generate a loop in the middle of a rerouted VP. This looping further deteriorates the performance of line restoration, although it is a relatively rare event compared to backhauling.

Cause 2. End-to-end restoration can geographically distribute the effect of failure over the network and lead to a higher level of sharing in spare capacity among failure events.

The more spare bandwidth is shared for restoration from different failure scenarios, the less spare capacity is needed in total. End-to-end restoration could attain a higher Degree of Sharing in Spare Capacity (DSSC) than line restoration since alternate paths could be dispersed in a geographically wider area. On the other hand, all alternate paths for the line restoration scheme must be terminated at the end-nodes of a failed link. Thus, required spare capacity would be more localized for each failure, and redundant bandwidth could be shared less among different link failure scenarios. For example, consider again a failure of link 2 and the restoration due to the LINE-OCA as well as the ETE-OCA (Figure 3-2). In order to recover a 400 BU[6] flow on arc 4-



(a) LINE-OCA

(b) ETE-OCA

Figure 3-2. Restoration flow of arc 4-1 upon a failure of link 2

All numbers in the figure are given in terms of BU (bandwidth unit).

6. BU (bandwidth unit) is an arbitrary unit of bandwidth, say Mbps.

1, the line restoration scheme requires a 400 BU spare capacity in total over the arcs leaving node 4. On the other hand, the end-to-end restoration scheme uses 100 BU less spare capacity in total since the affected path 5-4-1 is rerouted over a path away from node 4 (5-3-2-1). This extra bandwidth prepared for line restoration turns out to be unnecessary for other failure scenarios, implying less efficient usage of the redundant capacity.

Cause 3. The end-to-end restoration scheme, especially when combined with JOA, can share spare capacity more wisely by reclaiming the working capacity of affected VP's for restoration purposes.

Generally speaking, the links adjacent to a vertex with node degree 2 usually require relatively large amounts of spare capacity for line restoration. Upon a failure of either link, all flow on the link must be restored over the other link. Since there is no other alternative, increasing a flow on such links requires twice as much capacity installation for its primary use as well as for the restoration. Therefore, the LINE-JOA scheme usually avoids routing a flow via such a '*trap*' node. In our example, no flow is routed through node 1, and only demand originating or terminating at the node is assigned over its adjacent links (Table 3-2). These links still require twice as much bandwidth as the total demand coming into and going out of the node. A large part of their spare capacity is typically prepared for the recovery from a failure of the adjacent link. Thus, spare capacity would be inefficiently allocated on the links adjacent to a *trap* node with line restoration.

By routing more flow over a *trap* node, the end-to-end restoration scheme could reduce the total capacity installation cost. Since the ETE scheme can reclaim the working capacity of affected virtual paths for restoration purposes, link capacity can be used for an original flow assignment as well as for restoration. This capacity sharing is especially useful at the links where a large amount of spare capacity is required, such as the links around a *trap* node. Let us explain the case with the example of the small network. With the ETE-JOA scheme, 30 percent of the <2,4> demand is conveyed over the path 2-1-4 without increasing the total capacity over links 1 and 2 (Tables 3-2 and 3-3). Upon a failure of link 1, this traffic is rerouted over 2-5-4, and its working capacity of 300 BU over arc 1-4 is released (Figure 3-3). A flow of 400 BU previously originated from node 1 towards node 2 is now restored over arc 1-4 by claiming the capacity which was occupied by the <2,4> demand. Thus, the required spare capacity on arc 1-4 is only 100 BU although a flow of 400 BU must be restored upon the failure. As a result of this routing

and restoration strategy, we can reduce the working capacity of commodity <2,4> over arc 2-4 by 300 BU.

Reclaiming of working capacity does not necessarily happen around a *trap* node. In the above example, the <2-4> commodity over path 2-1-4 is switched to 2-5-4 upon a failure of link 1. However, only 200 BU spare capacity is allocated over arc 2-5 while the rerouted bandwidth amounts to 300 BU. An additional 100 BU for the restoration comes from the working capacity which was occupied by the <1,5> demand over 1-2-5 and released upon the failure. The reclaiming effect is further observed in the <4,3> commodity where a third of its demand is routed over a longer path. In summary, the ETE-JOA approach can realize economical capacity allocation by using part of a link capacity for both an original flow assignment and a restoration from a failure scenario.

The JOA can save resource installation cost over the OCA for the following reason:

Cause 4. The JOA scheme can adjust a flow so that the spare capacity can be efficiently shared to reduce the total capacity installation cost.

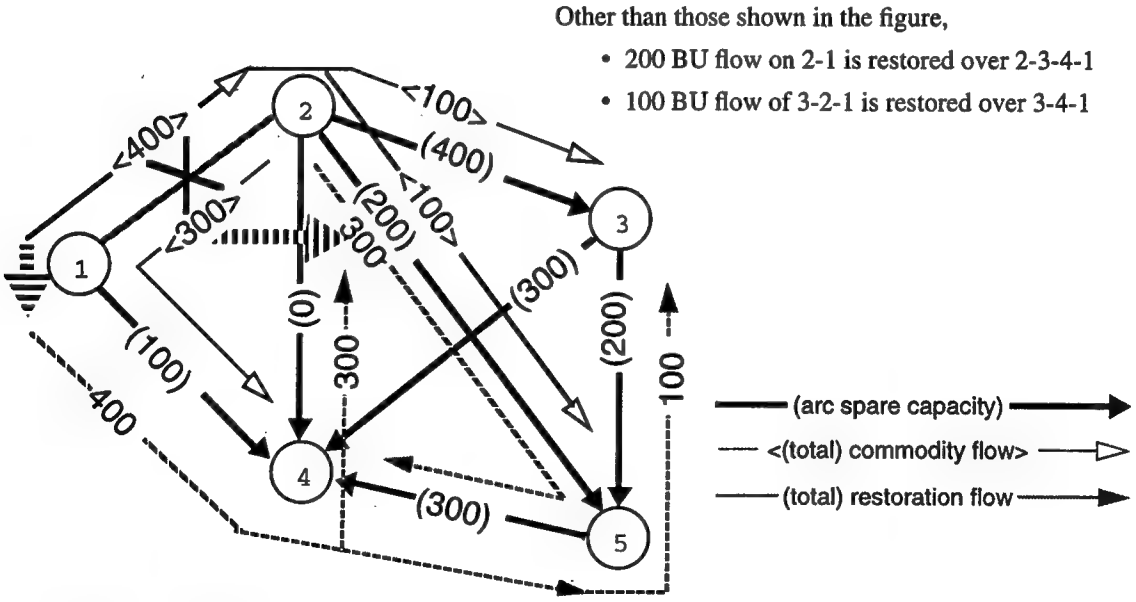


Figure 3-3. Restoration flow upon a failure of link 1

The ETE-JOA scheme is employed in this example. All numbers in the figure are given in terms of BU (bandwidth unit).

It is worthwhile to note that the minimum working capacity cost[7] is attained by the flow assignment due to the OCA since the shortest route flow is used. The JOA requires at least equal and typically more working capacity cost. By sharing the spare capacity more efficiently, however, the JOA could attain reduction on the total capacity cost[8]. For example, compare the results of the LINE-JOA and the LINE-OCA. The <1,5> demand is routed over 1-2-5 with the JOA rather than its shortest path 1-4-5 (Table 3-2). Although this results in 50 CU[9] more working arc cost, the spare cost of arc 1-4 is reduced by 100 CU, leading to a cost reduction of 50 CU (Table 3-3). Due to the relatively large demand for <2,4>, more redundant bandwidth is installed over 2-1-4 than 4-1-2. Thus, it is beneficial to adjust a flow over links 1 and 2 so that more spare capacity would be required over 2-1-4 than 4-1-2 upon a failure of the links. Thus, the JOA assigns the <1,5> commodity over 1-2-5 instead of 1-4-5.

3.4.2. Effect of network topology

Further experiments have been carried out to analyze the effectiveness of the end-to-end restoration and joint optimization in a wide variety of network environments. Eight sample networks with diverse topological characteristics have been explored in the experiment. Four of them are real networks which had appeared in the literatures [24] [35] [80] [83] (Figure 3-4). The other four are artificial networks (Figure 3-5). In the experiments, two commodities are assumed to be defined between any node pair, one for each direction. The following projected traffic demand patterns have been employed: A uniform (UF) demand pattern with 1,000 BU between any node pair, and a weighted (WT) demand pattern with 1,000 BU between any adjacent nodes and 500 BU for the others[10]. Furthermore, random (RD) demand patterns, where the demand of each commodity is uniformly distributed between 250 BU and 1,750 BU are considered. The JOA and OCA are investigated for each restoration scheme.

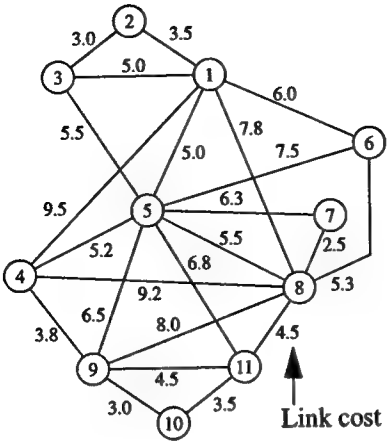
The result is expressed in terms of a normalized spare capacity cost (*NC*), which is defined as follows: Let D^* be the minimum link cost required to support the traffic demand. Obviously, the

7. The working capacity cost is defined to be the cost of the capacity required to satisfy the traffic demand. This cost is determined by a commodity flow assignment.

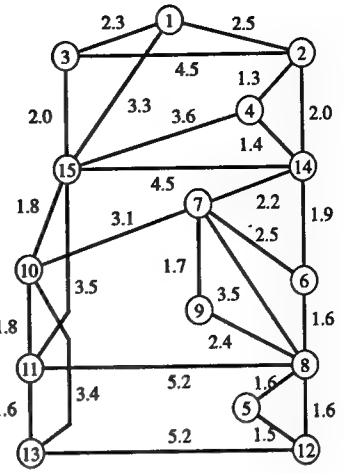
8. The total capacity cost for a fully restorable network is given as the sum of the working capacity cost and the cost of the spare capacity.

9. CU (cost unit) is an arbitrary unit of arc capacity installation cost. In this example, an arc installation cost per one BU is given in terms of CU (refer to Table 3-3).

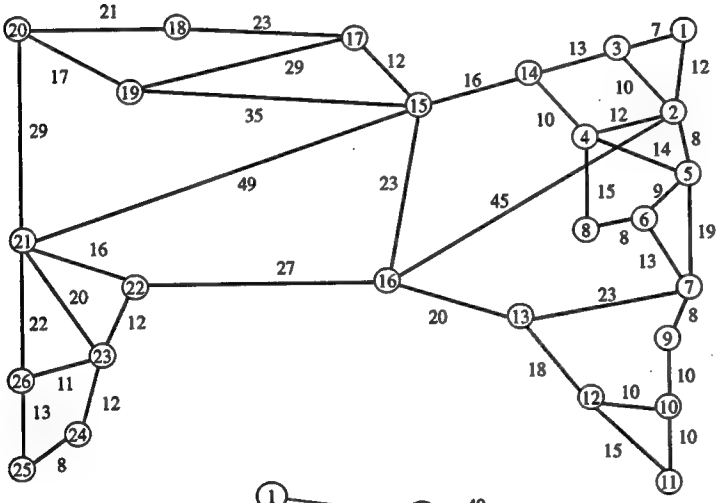
10. A weighted demand pattern comes from the assumption that more traffic would be generated between adjacent nodes than others.



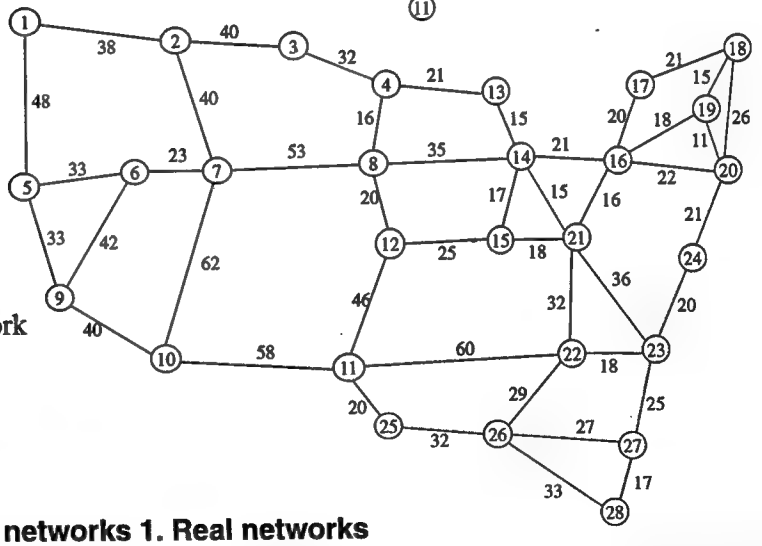
(a) NJ-LATA network , (11,46)
DA=2.57, DE=2.93



(b) LATA network
DA=2.25, DE=2.96
(15,56)



(c) ARPA network
DA=1.73
DE=2.41
(26,82)

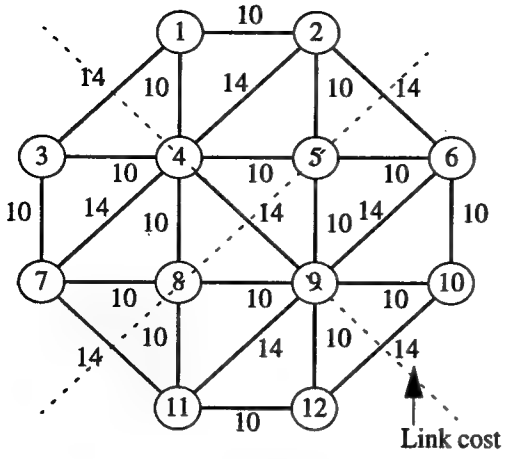


(d) US-WAN network
DA=1.87
DE=2.60
(28,90)

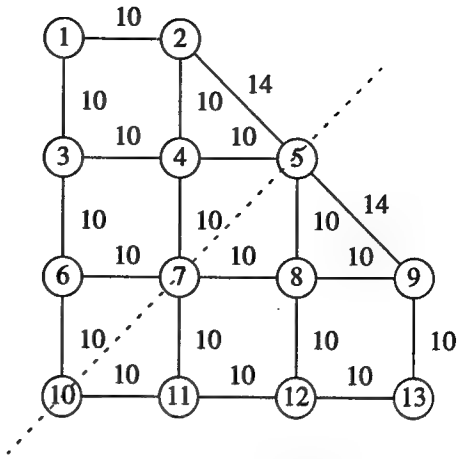
Figure 3-4. Sample networks 1. Real networks

DA = The average number of disjoint arc restoration paths

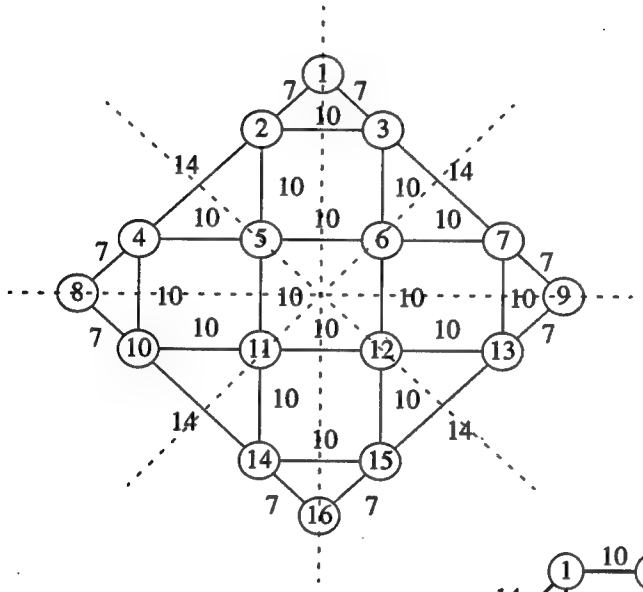
DE = The average number of disjoint end-to-end paths
Refer to page 49, for their definitions.



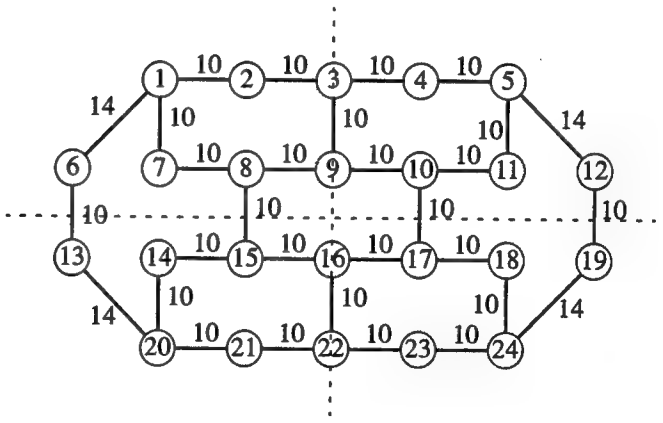
(e) (12,50) network
DA=2.72, DE=3.47



(f) (13,40) network
DA=1.9, DE=2.65



(g) (16,56) network
DA=2.43, DE=3.1



(h) (24,60) network
DA=1.27, DE=2.24

Figure 3-5. Sample networks 2. Artificial networks

The dotted lines represent the axes of symmetry.

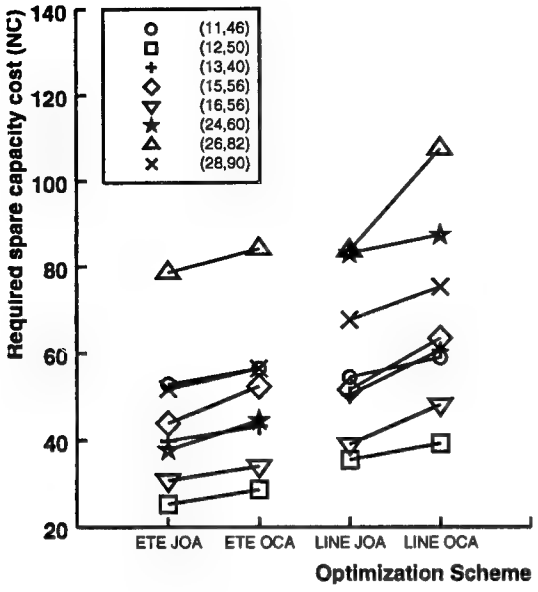
DA = The average number of disjoint arc restoration paths
DE = The average number of disjoint end-to-end paths

minimum is attained when all demands are sent over their shortest routes. Let D_T be the total link cost obtained using a fully restorable capacity placement scheme *T*. Now, define the required spare capacity cost of T as an additional cost necessary to realize full restorability with the scheme, which is given by $(D_T - D^*)$. The normalized spare capacity cost (*NC*) is obtained through the normalization of the above cost with respect to D^*, which is given by $100 \cdot (D_T - D^*)/D^*$ %. Note that the additional cost is not necessarily equal to the cost of the actual spare bandwidth because a flow assignment may be different from a shortest route flow in case of joint-optimization. Figure 3-6 summarizes the required spare capacity cost for each optimization scheme.

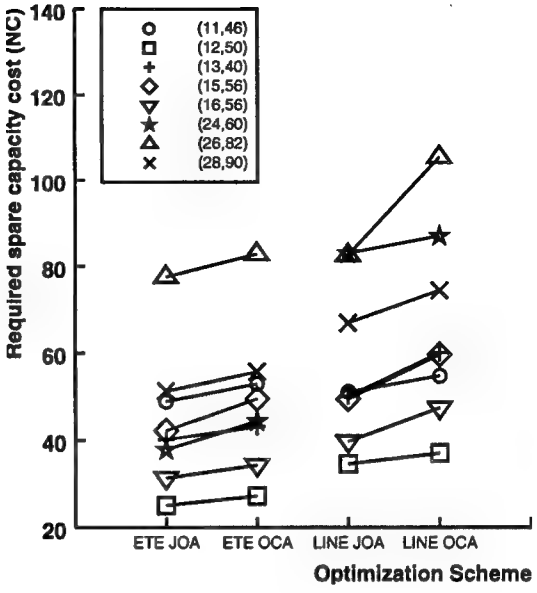
First of all, the demand pattern does not have a significant influence on the effectiveness of each optimization scheme (Figure 3-6-a~d). As expected, the JOA requires less spare capacity than the OCA, and the end-to-end restoration outperforms the line restoration in terms of spare capacity cost. The improvement ratio appears to be almost consistent among different demand patterns, although the required spare capacity cost differs accordingly. On the other hand, network topology turns out to have a significant impact on the required spare capacity cost. In particular, the degree of improvement due to end-to-end restoration over line restoration depends considerably on the network model (Figure 3-6-e), while the benefit due to joint optimization (JOA over OCA) turns out to be less sensitive to the network topology (Figure 3-6-a).

Then what kind of network topology obtains a more economical benefit by using an end-to-end restoration scheme? Before answering this question, we first consider the three major advantages of the end-to-end restoration scheme discussed in Section 3.4.1 and examine their effect on the total redundant capacity cost. This analysis is used for further discussion about the effect of network topology later. The eight sample networks with the UF and WT traffic demands are employed in the experiments.

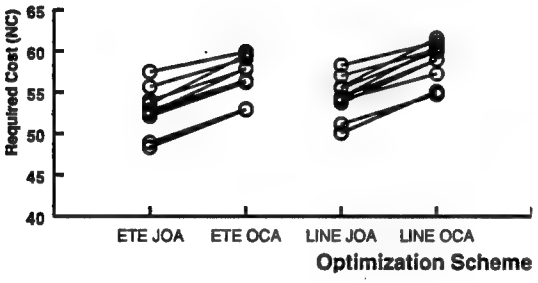
First of all, we investigate the effect of geographical distribution of alternate routes on the required spare capacity cost. As discussed in the previous section, the ETE scheme is expected to share more spare capacity among different failure scenarios since the restoration paths could be widely distributed (Cause 2). Thus, the scheme could reduce the spare capacity installation. In the experiment, the degree of sharing in spare capacity (DSSC) is measured by the average spare capacity usage per failure, *W*, which is given by:



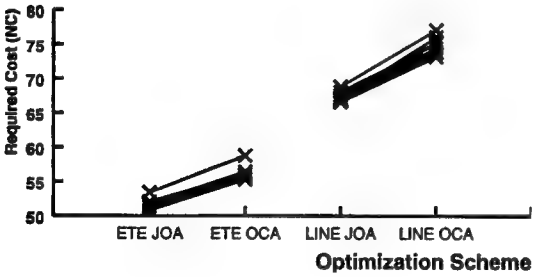
(a) UF demand



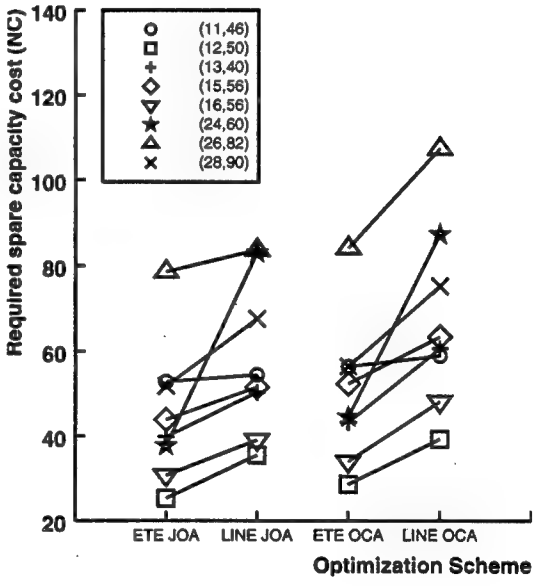
(b) WT demand



(c) NJ-LATA - UF+WT+8 RD's



(d) US-WAN - UF + WT+ 8 RD's



(e) UF demand

Figure 3-6. Required spare capacity cost for each scheme

Required spare capacity cost is given in terms of a normalized spare capacity cost (*NC*). The figures (a) and (e) show the same result, but the orders of the optimization schemes in the x-axis are different. The former is intended to emphasize the difference by the JOA and OCA schemes, while the latter attempts to contrast the ETE and LINE schemes.

$$W = \sum_{l \in E} W_l / |E|$$

$$W_l = \sum_{a \in A_l} d_a \cdot \left(\max \{ (z_a^l - r_a^l), 0 \} \right) / \sum_{a \in A_l} d_a \cdot z_a$$

where $A_l \equiv A \setminus \{a : a \in l\}$. Note that r_a^l is zero with line restoration. W_l gives a fraction of the total spare capacity (weighted by arc cost) which is actually consumed upon a failure of link l. W is the average of W_l over all failure scenarios.

Figure 3-7 depicts the relationship between W and the normalized spare capacity cost for the ETE and LINE schemes. The line in the figure is the least-square line over all sample points. The result indicates that the required spare capacity cost is largely determined by the attainable DSSC for end-to-end restoration. Such a relationship also holds for the line restoration, but it has a few remarkable exceptions. The (24,60) network requires considerably more spare bandwidth compared to their attainable DSSC, while the NJ-LATA network needs relatively less spare bandwidth.

Backhauling and looping are identified as other factors that cause the inefficient use of spare bandwidth in the line restoration scheme (Cause 1 in the previous section). We next investigate the effect of backhauling and looping on the required spare bandwidth for the line restoration-based networks. The effect is measured by the average wasted spare bandwidth cost per failure due to backhauling and looping, BL. It is defined by:

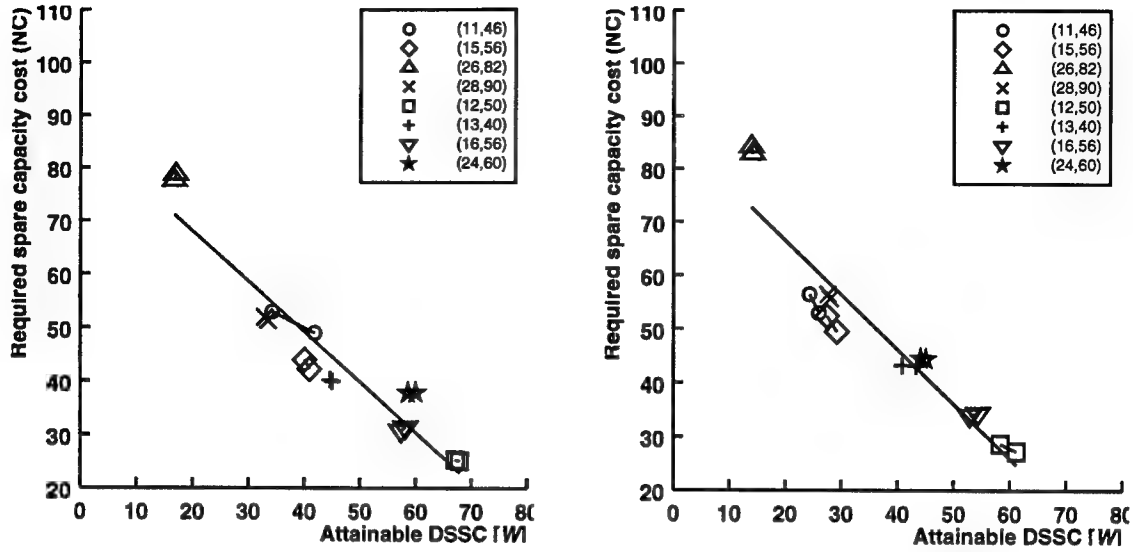
$$BL \equiv \sum_{l \in E} BL_l / |E|$$

$$BL_l \equiv \left(\sum_{a \in A_l} d_a \cdot bl_a^l \right) / \left(\sum_{a \in A_l} d_a \cdot z_a^l \right)$$

where bl_a^l is the wasted bandwidth of arc a due to backhauling or looping upon line restoration from a failure of link l[11]. BL_l is a fraction of the wasted spare bandwidth (weighted by arc cost) over the bandwidth actually used for the restoration of link l. BL is the average over all single-link failures.

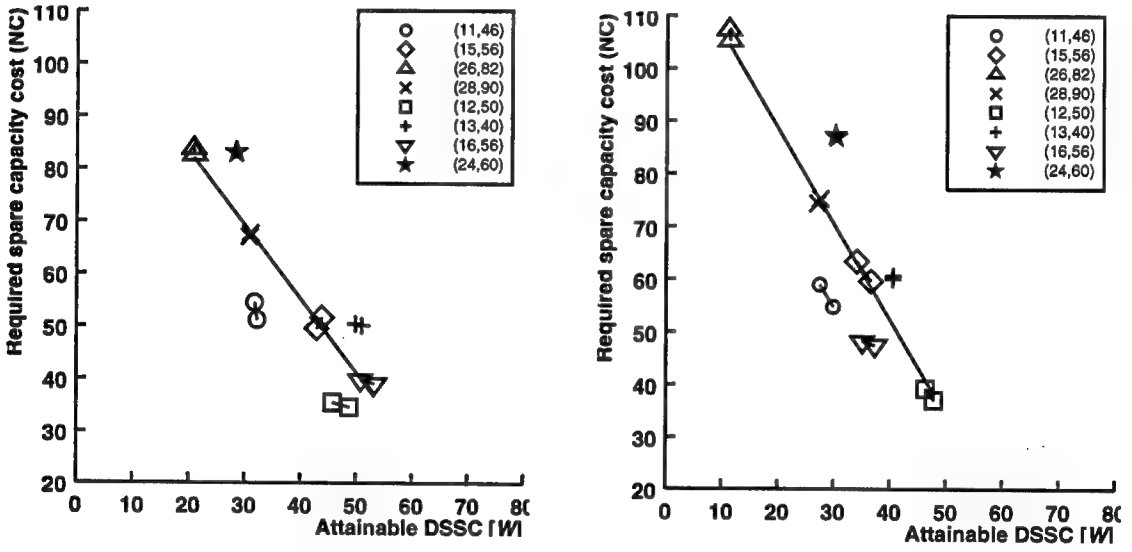
---

11. The amount of wasted bandwidth depends on how the virtual paths are restored over the restoration paths. In the example of Figure 3-2-(a), a virtual path 5-4-1 results in backhauling if it is restored over 5-4-5-3-2-1, while the path does not suffer from backhauling if it is rerouted over 5-4-2-1. The output of the SCFA-MF-LINE problem, however, does not specify which virtual paths are rerouted over which restoration paths. In the calculation of bl_a^l, it is assumed that the bandwidth of the affected traffic is uniformly distributed over the restoration paths.



(a) MF-ETE / JOA

(b) MF-ETE / OCA



(c) MF-LINE / JOA

(d) MF-LINE / OCA

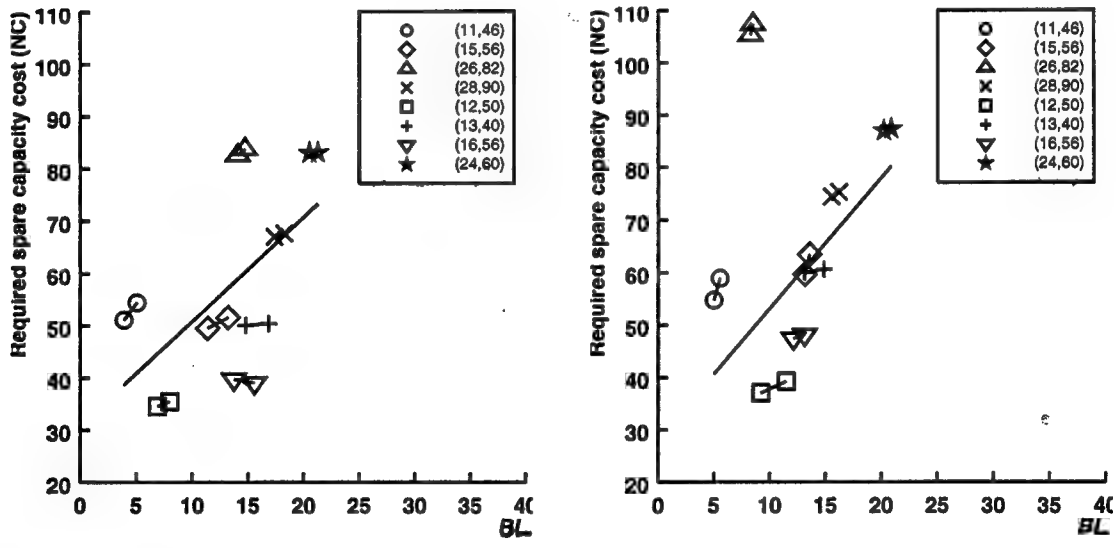
Figure 3-7. The effect of attainable DSSC on required spare capacity cost

Two demand patterns, UF and WT, are utilized for each sample network. The spare capacity cost is expressed in terms of a normalized spare capacity cost (*NC*). The line in the figures is the least-square line over all samples. The closer all sample points are to the line, the closer the relationship observed between required spare capacity cost and attainable DSSC.

Figure 3-8 illustrates the effect of backhauling and looping on the required spare capacity cost. The line in the figure is the least-square line over all sample points. The result shows that the degree of backhauling and looping somewhat varies by network topology, and it has a very loose relationship to the required spare capacity cost. The wasted bandwidth [*BL*] is nominal in the NJ-LATA network, which accounts for the lower spare capacity requirement compared to its attainable DSSC in Figure 3-7-(c,d). On the other hand, the (24,60) network ends up with the most waste, requiring a relatively higher spare capacity installation as in Figure 3-7-(c,d).

The end-to-end restoration scheme can further reduce the spare capacity cost by reclaiming the bandwidth of affected VP's for restoration purpose (Cause 3 in the previous section). The average reclaimed spare bandwidth cost per failure, *RC*, is explored to clarify its dependency on the required spare capacity cost. *RC* is given by:

$$RC \equiv \sum_{l \in E} RC_l / |E|$$



(a) MF-LINE / JOA (b) MF-LINE / OCA

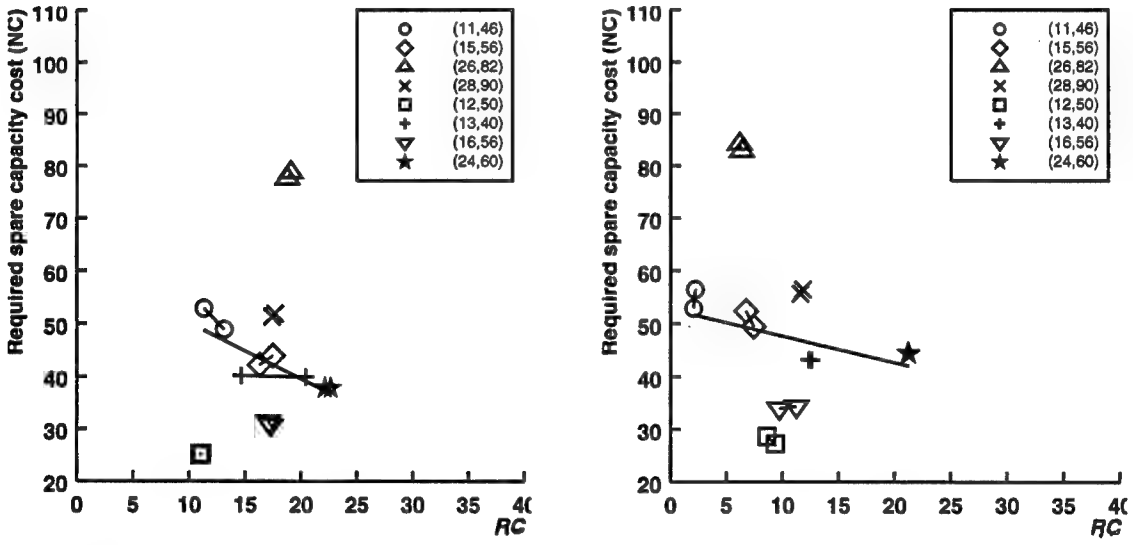
Figure 3-8. The effect of backhauling and looping on required spare capacity cost

Two demand patterns, UF and WT, are employed for each sample network. The spare capacity cost is expressed in terms of a normalized spare capacity cost (*NC*). The line in the figures is the least-square line over all samples.

$$RC_l \equiv \sum_{a \in A_l} d_a \cdot r'^{l}_{a} / \sum_{a \in A_l} d_a \cdot z^{l}_{a}$$

where r'^{l}_{a} is the total amount of reclaimed bandwidth of arc a upon a failure of link l. The bandwidth released and reclaimed by the same affected VP is excluded in this calculation[12]. Thus, r'^{l}_{a} is not equal to r^{l}_{a}. RC_l is the percentage of the reclaimed bandwidth (weighted by arc cost) over the total bandwidth spent for the restoration from a failure of link l, and RC is its average over all failure scenarios.

Figure 3-9 shows the result of this experiment. The required spare capacity cost is plotted against RC. The result indicates that the effect of bandwidth reclaiming does not appear to have any explicit relationship to the required spare capacity cost. This suggests that it is not a ruling factor for the effectiveness of the end-to-end restoration scheme.



(a) MF-ETE / JOA (b) MF-ETE / OCA

Figure 3-9. The effect of bandwidth reclaiming on required spare capacity cost

Two demand patterns, UF and WT, are employed for each sample network. The spare capacity cost is expressed in terms of a normalized spare capacity cost (*NC*). The line in the figures is the least-square line over all samples.

12. For example, suppose that a virtual path 5-4-1 in Figure 3-1 is rerouted over a restoration path 5-4-2-1 upon a failure of link 2. Upon restoration, the bandwidth over arc 5-4 is released and reclaimed by this virtual path. Since this type of bandwidth reclaiming by the ETE scheme does not give any advantage over the LINE scheme, the amount of the bandwidth reclaimed in this manner is excluded from the calculation of r'^{l}_{a}.

In summary, the degree of sharing in spare capacity (DSSC) turns out to be a major determining factor of spare capacity cost, while the backhauling and looping effect also contributes to its determination for networks based on the line restoration scheme.

Next, we examine how network topology influences these factors and determines the required spare capacity cost. A network can be characterized by the two topological aspects: *connectivity* and *connection regularity*. We find that these two measures can reflect the effect of the two key factors in determining the redundant capacity cost: the DSSC and the backhauling. Connectivity expresses how well nodes are connected in a network. It is expected that a well-connected network requires less spare capacity than a sparse network, since the former would have more candidate restoration paths than the latter. In the following discussion, the average number of disjoint arc restoration paths, DA, is employed as the measure of the connectivity for the networks based on the line restoration scheme[13]. It is defined by:

$$DA = \sum_{a \in A} DA_a / |A|$$

where DA_a is the number of disjoint (i.e. mutually exclusive) restoration paths for arc a. The more disjoint restoration paths, the higher the DSSC that can be attained, since the effect of failure can be distributed over more restoration paths. As for the network based on the end-to-end restoration scheme, the average number of disjoint end-to-end paths, DE, is employed as the measure of the connectivity. It is defined by:

$$DE = \sum_{\pi \in \Pi} DE_\pi / |\Pi|$$

where DE_π is the number of disjoint paths between the source and destination nodes of commodity π.

Connection regularity is another important topological measure. The measure should express how evenly the links are connected over a network and how symmetrically the connection is arranged in a network. If a connection is well-balanced, it is expected that the affected VP's can be distributed over the network more evenly, which could promote attainable DSSC among different

13. An average node degree can also be used for the connectivity measure. A degree of node v is defined to be the number of links incident with v. An average node degree is its average over all nodes in a network. However, we find that DA is a more suitable metric for the network connectivity than the average node degree because the former represents a global connectivity between the end-nodes of a failed link, while the latter only captures a local view at a node.

failure scenarios. A major obstacle here is that the measure is very hard to quantify. In the following discussion, we classify eight sample networks into two categories: balanced networks and unbalanced networks. A good measure for the connection regularity needs to be considered further in future works. The eight sample networks can be categorized as follows:

1. Balanced networks: (12,50), (16,56), (13,40) and (24,60) networks in decreasing order of network connectivity.
2. Unbalanced networks: NJ-LATA, LATA, US-WAN, and ARPA networks in decreasing order of network connectivity.

First consider the case of end-to-end restoration. Figure 3-10-a,b depicts the relationship between connectivity (*DE*) and the required spare capacity cost. The solid line in the figure is the least-square line over all sample points, while the dashed line is the least-square line over all samples of the balanced networks, and the dotted line is that over all samples of the unbalanced networks. The figure shows that all balanced sample networks can require less spare capacity than any unbalanced model networks[14]. For example, the (24,60) network requires less spare bandwidth than the LATA network, although the former has a lower level of network connectivity than the latter. This indicates that connection regularity is a major factor in determining the required spare capacity.

Among the balanced networks, well-connected networks can generally require less spare capacity cost. Since a network with a higher connectivity has more disjoint restoration paths, it can distribute the effect of a failure to a wider area, resulting in a higher DSSC. The only exception is the (13,40) network, which requires a slightly higher amount of redundant capacity than the (24,60) network in spite of its higher connectivity. This is because the (13,40) network is less symmetrical than the (24,60) network[15]. This suggests that regularity has a significantly larger impact on the required capacity. The same argument on the effect of connectivity can apply for the unbalanced networks, but it holds only weakly. A possible cause is the difference in the regularity level among those networks. In summary, connection regularity turns out to be an important factor in deciding the redundant capacity cost of fully restorable networks with end-to-end restoration, while connectivity plays a minor role in the determination of cost.

---

14. As shown in Figure 3-7-(a,b), all balanced networks can attain a higher DSSC than any unbalanced network.

15. The (13,40) network is symmetric along only one axis, while the (24,60) network is symmetric along two axis (see the dotted lines representing the axes of symmetry in Figure 3-5).

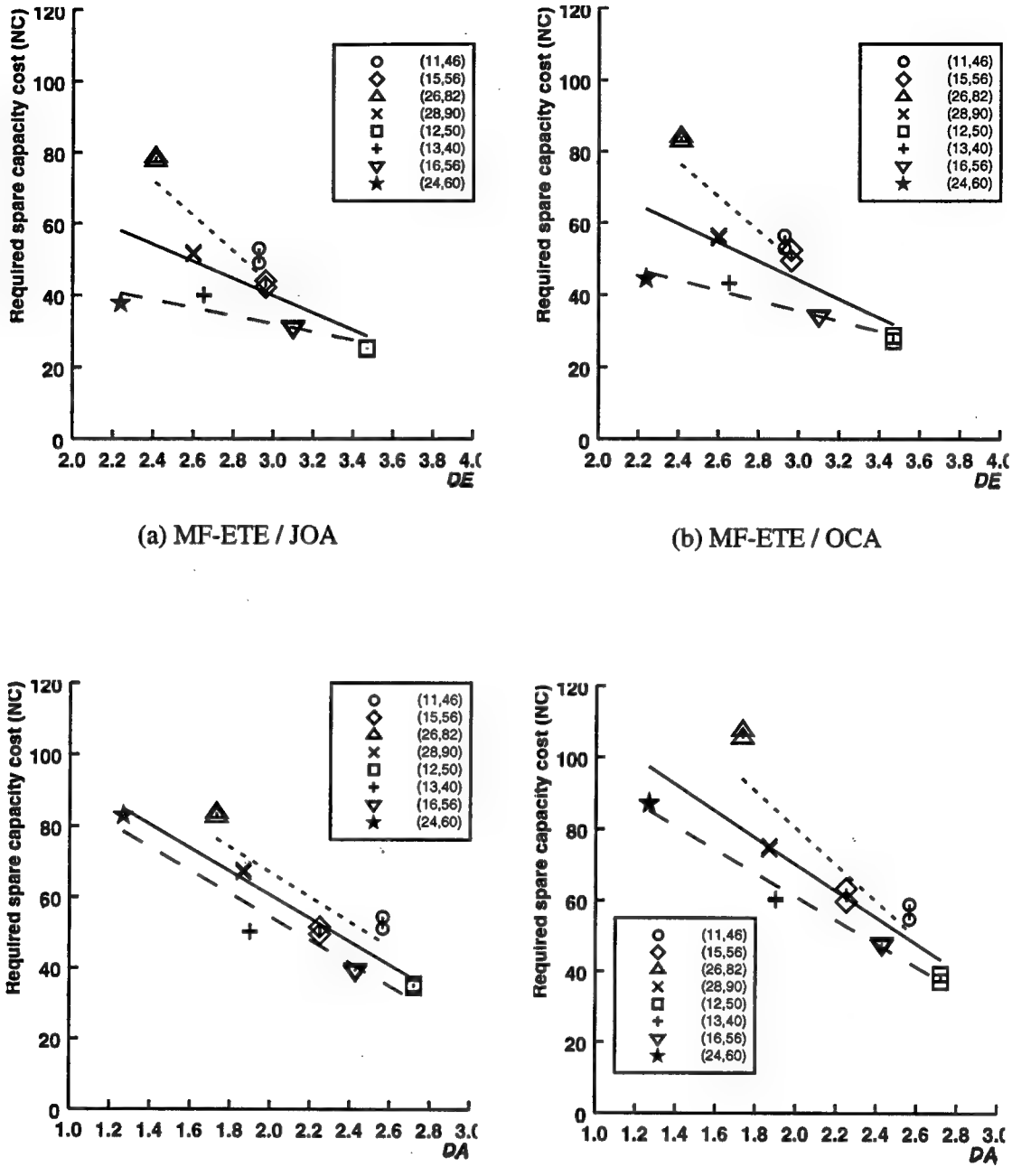
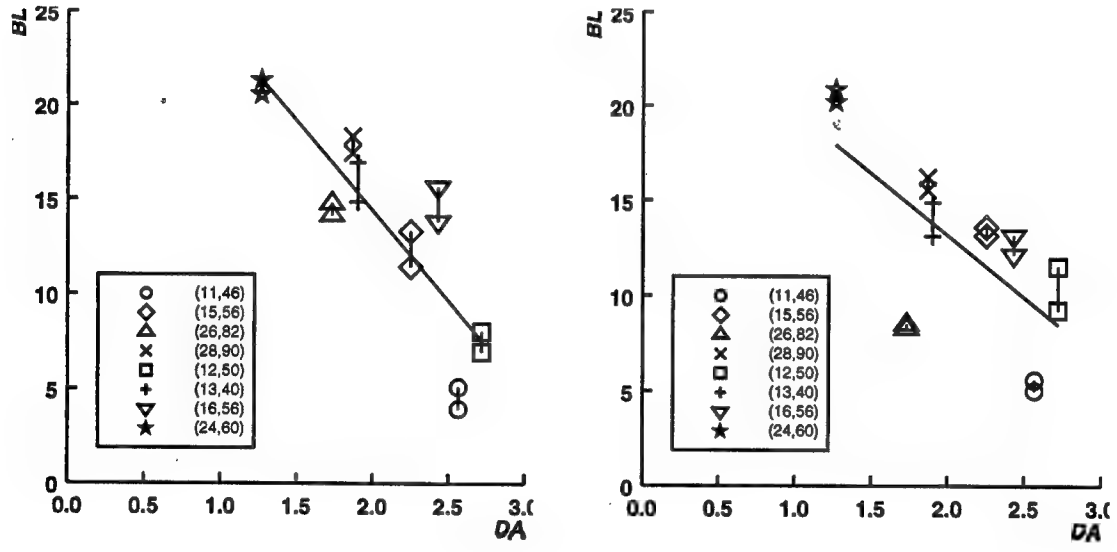


Figure 3-10. Topological effect 1. Effect of Connectivity

Two demand patterns, UF and WT, are employed for each sample network. The spare capacity cost is expressed in terms of a normalized spare capacity cost (*NC*). The solid line in the figures is the least-square line over all samples, while the dashed line is the least-square line over all samples for the balanced networks, and the dotted line is that over all samples for the unbalanced networks.

Next, consider the case of line restoration (Figure 3-10-c,d). The figure shows that balanced networks require less spare capacity than unbalanced networks if they have a similar level of connectivity. For example, although the connectivity level (*DA*) is almost the same for the (13,40) and US-WAN networks, the former requires less spare bandwidth due to its connection regularity. Unlike the end-to-end restoration, however, connection regularity is not a single key factor but connectivity also plays an important role in determining the necessary spare capacity cost. For example, the (24,60) network requires a considerable amount of spare bandwidth due to its sparseness even though it is well-balanced. Since rerouting is executed between the two end-nodes of an affected link, all restoration flow must be sent over the unaffected arcs adjacent to the nodes. However, the number of such arcs is very small for a sparse network. Consequently, a larger amount of spare capacity must be installed over the arcs. Furthermore, backhauling happens with a higher possibility for a network with a low connectivity. A VP relayed at a node adjacent to a failed link is more likely to be rerouted back to the same link it traversed. This assertion is verified in Figure 3-11. In summary, both connection regularity and connectivity are the key factors in determining the necessary spare bandwidth for fully restorable networks based on line restoration.



(a) MF-LINE / JOA

(b) MF-LINE / OCA

Figure 3-11. Topological effect 2. Backhauling and connectivity

The figure shows the relationship between the backhauling and looping effect (*BL*) and connectivity (*DA*). Two demand patterns, UF and WT, are employed for each sample network. The line in the figures is the least-square line over all samples.

Now we address the following question posed before: What kind of network topology gets more economical benefit from the end-to-end restoration scheme? The cost savings obtained by end-to-end restoration over line restoration are summarized in Table 3-3. The savings rate ranges widely from 3 to 55 percent, suggesting that network topology is a crucial factor for economical gain from end-to-end restoration. In general, large cost savings are expected for balanced sparse networks, as in the (24,60) network. This is because regularity gets a significant advantage from end-to-end restoration, and sparsity places a considerable penalty on line restoration. On the other hand, such a disadvantage posed on line restoration may fade away for well-connected networks. Furthermore, end-to-end restoration may not be able to reap a large gain for an unbalanced network. Consequently, the cost savings may be small for unbalanced and/or well-connected networks[16]. This accounts for the fact that only nominal savings can be obtained in the NJ-LATA (unbalanced and well connected) network, and small savings can be obtained for the ARPA (unbalanced) network. In such networks, it may not be advisable to employ the end-to-end restoration scheme, considering its complicated decision making process on the alternate routes.

Finally, the cost savings with joint optimization are summarized in Table 3-4. Figure 3-12 com-

Table 3-3. Cost savings of ETE-JOA over LINE-JOA[a]

Network model	Unbalanced Networks				Balanced Networks			
	Well-connected <--------> Sparse				Well-connected <--------> Sparse			
	NJ-LATA	**LATA**	**US-WAN**	**ARPA**	**(12,50)**	**(16,56)**	**(13,40)**	**(24,60)**
Cost (ETE)[b]	52.86	43.93	51.86	78.69	25.31	30.76	39.86	37.80
Cost (LINE)	54.45	51.60	67.65	83.71	35.52	39.07	50.44	83.07
Savings rate[c]	2.93	14.87	23.35	6.00	28.74	21.27	20.98	54.50
Cost savings[d]	1.59	7.67	15.79	5.02	10.21	8.31	10.58	45.27

a. UF demand pattern is used in the experiment.
b. (NC)
c. {LINE-ETE} / {LINE} (percent)
d. {LINE-ETE} (NC)

16. This is the contrapositive of the previous assertion that "large cost savings can be attained for balanced and sparse networks". The contrapositive of the proposition '$p \rightarrow q$' is given by '$\neg q \rightarrow \neg p$'. The contrapositive is true if the original proposition is true. Now, the contrapositive of the above assertion can be stated as "if there is a network with nominal cost savings, then it happens for unbalanced and/or well-connected networks". The results of the NJ-LATA sample network suggest the existence of such a network.

pares the normalized spare capacity cost of the JOA and OCA schemes. As discussed before, the comparative study with the OCA scheme elucidates the advantage of the joint optimization. The result indicates that on average over 10 percent of the cost can be saved by the JOA over the OCA. The cost savings ratio ranges from 6 to 22 percent. A less drastic dependency on network topology is observed compared with the savings of the ETE over the LINE.

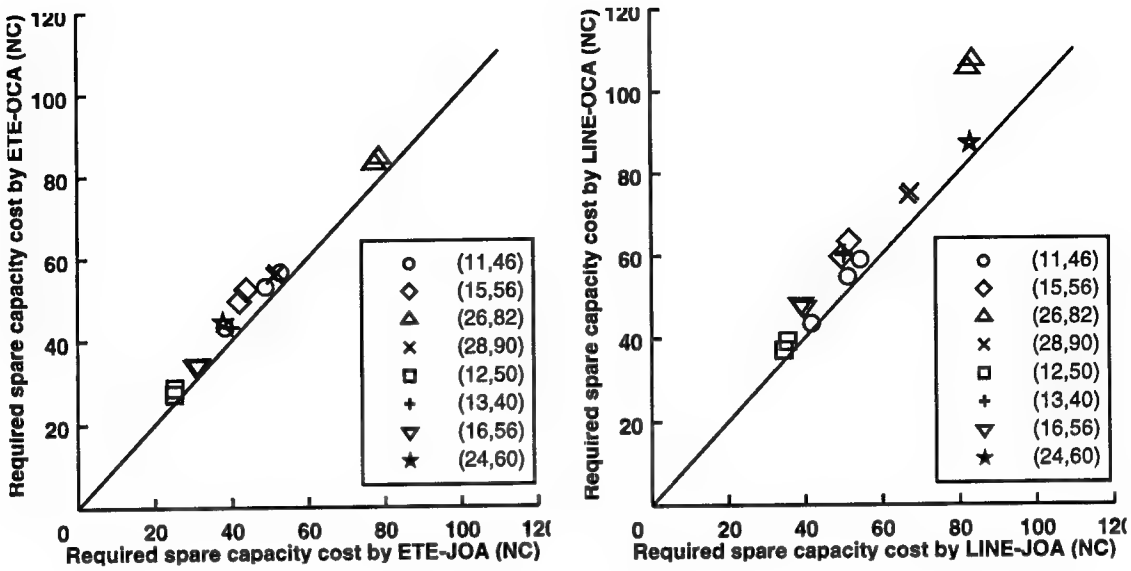
3.5. Summary

Two main results in this section are the development of joint optimization methods for the SCFA problem and a comparative analysis of the restoration schemes with respect to required spare capacity cost. Previously, no work has been reported on a joint optimization approach for self-healing networks. Since the optimization is conducted jointly over capacity assignment, commodity flow assignment and restoration flow assignment, however, the complexity of the problem grows tremendously. In order to make the problem computationally tractable, several mechanisms are developed. *LU* factorization of the basis matrix is facilitated by exploiting its special structure with the low density of non-zero elements. A direct *LU* update technique is proposed to further economize the computation. A row generation and deletion mechanism is equipped to cope with the explosive number of constraints in the SCFA-MF-ETE problem. The economical benefit of the end-to-end restoration scheme is thoroughly examined with respect to the globally optimal capacity installation cost. The degree of sharing in spare capacity (DSSC) appears to be a key determin-

Table 3-4. Cost savings of JOA over OCA

Network Model		NJ-LATA	LATA	ARPA	US-WAN	(12,50)	(13,40)	(16,56)	(24,60)	Average
MF-ETE	Cost (JOA)[a]	52.86	43.93	78.69	51.86	25.31	39.86	30.76	37.80	
	Cost (OCA)	56.46	52.39	84.18	56.49	28.66	43.26	34.00	44.60	
	Savings rate[b]	6.39	16.16	6.53	8.19	11.69	7.87	9.54	15.25	10.20
MF-LINE	Cost (JOA)	54.45	51.60	83.71	67.65	35.52	50.44	39.07	83.07	
	Cost (OCA)	58.96	63.44	107.4	75.30	39.28	60.65	48.17	87.37	
	Savings rate	7.65	18.66	22.04	10.15	9.57	16.85	18.89	4.92	13.59

a. Required spare capacity cost in terms of NC (Normalized cost)
Cost is normalized by the minimum average link capacity satisfying the demand.
b. {OCA-JOA} / {OCA} (percent)
The UF demand pattern is employed in the experiment.



(a) MF-ETE

(b) MF-LINE

Figure 3-12. Required spare capacity cost: JOA vs. OCA

ing factor of spare capacity cost, while the backhauling and looping effects also contribute to it for line restoration-based networks. The result suggests that a balanced sparse network can gain a remarkable benefit from end-to-end restoration. Contrary to a wide belief in the economic advantage of the end-to-end restoration scheme, the analysis has revealed that the attainable gain could be marginal for a well-connected and/or unbalanced network.

IV Survivable Virtual Path Routing (SVPR)

The SVPR problem arises when there is a major change in the underlying network environment such as a traffic demand or a facility network reconfiguration. The VP planner performs dynamic VP reconfiguration to maximize network survivability in a new environment. Three restoration mechanisms discussed in Section 2.1.2 are considered: MF-LINE, MF-ETE and KSP-LINE. Due to substantial differences in their restoration mechanisms, their optimization models and solution procedures are developed independently. The SVPR problems for MF-LINE or MF-ETE networks are modeled as a large-scale LP problem as in the SCFA problem, and similar strategies are used to solve these problems. On the other hand, the SVPR problem for KSP-LINE networks is modeled as a nonlinear, non-smooth multicommodity flow problem with linear constraints. New algorithms are developed to find a near-optimal solution. The effectiveness of the proposed dynamic VP routing scheme is examined through numerical experiments. A comparative analysis of the three restoration schemes clarifies their pros and cons with respect to network restorability. This section further demonstrates the efficiency of the two-step restoration scheme proposed in Section II.

4.0. Introduction

The SVPR minimizes an expected amount of lost flow upon restoration from a network failure. It aims to enhance the survivability of the network in response to a change in the network environment. Previous works on fast restoration schemes have overlooked the need of dynamic VP reconfiguration and have attempted to meet a restorability objective through the development of restoration protocols. They assume that a flow assignment is static. However, a static virtual path routing approach not only degrades the performance of a fast restoration protocol, but it also requires more spare capacity to achieve the same level of survivability as in a dynamically reconfigurable network. Therefore, a dynamic VP reconfiguration scheme, combined with an optimal network design scheme proposed in the previous section, can realize a self-healing ATM network efficiently and in a cost-effective manner.

The rest of this section is organized as follows. Section 4.1 gives the formal definition of the SVPR problem. The succeeding three sections discuss the details of the SVPR problems based on three different restoration schemes. The problem formulation and solution approach are discussed in the respective sections. Finally, results from numerical experiments are reported in Section 4.5.

4.1. Formal Problem Definition

Using the same notations[1] and assumptions introduced in Section 3.1, the SVPR problem can be formally stated as the following optimization problem:

Given	$G = (V, A, \underline{c})$ and Q,		**(SVPR)**
Minimize	An expected lost flow $L(\underline{f})$		
over	$\underline{f}, \underline{r}$		
subject to	a) $\underline{f}$ is a multicommodity flow satisfying Q,	(flow conservation law)	
	b) capacity constraints are satisfied, and	(capacity constraints)	
	c) restoration flow $\underline{r}$ obeys the fast restoration protocol	(restoration flow constraints)	

As in the SCFA problem, three fast restoration schemes lead to three distinct SVPR problems. They are (1) MF rerouting with line restoration, (2) MF rerouting with end-to-end restoration and (3) KSP rerouting with line restoration. In the following, the corresponding SVPR problems are referred to as the SVPR-MF-LINE, SVPR-MF-ETE and SVPR-KSP-LINE problems, respectively.

4.2. SVPR-MF-LINE

4.2.1. Problem formulation

The intricate math is shown in Appendix H in order to preserve the flow of the document.

1. Also refer to Appendix T.

4.2.2. Solution approach

The intricate math is shown in Appendix I in order to preserve the flow of the document.

4.2.3. Validity of the algorithm

The intricate math is shown in Appendix J in order to preserve the flow of the document.

4.2.4. Implementation issue

As discussed in Section 2.2.4, the speed of the VP planner module determines a feasible range of an update interval of VP configuration. The VP planner should output the result as quickly as possible, say in a period one or two orders of magnitude less than the VP update interval. In order to clarify the applicability of the proposed algorithm, this section examines the required CPU time for four sample networks. The experiments are conducted on the DEC Alpha-station 200 4/233. The uniform traffic (UF) demand and its jointly optimal capacity placement are utilized in the experiments. The traffic demand projected at the design phase of the capacity assignment is called a *base traffic demand* in this project. A *relative network load* (RNL) is used as a metric of the network load. It is defined to be 100% if the offered traffic load is the same as the base traffic demand.

The CPU time is measured at 100% RNL as well as 102.5% RNL. Physical resource allocation is designed based on demands of the busy hour traffic projected over some future time frame. Thus, the demand with 100% or less RNL represents the case when the network is in a normal operating condition. On the other hand, 102.5% RNL represents the case when there is an occasional surge of traffic demand beyond the projected level.

For the latter case, the base traffic demand is uniformly increased by 2.5 percent. At 100% RNL, on the other hand, 30 randomly generated traffic demands are utilized[2]. The end-to-end traffic requirements of each random input deviate at most 10 percent from those of the base traffic demand. They are uniformly distributed over this range, while the total amount of traffic demand is fixed. A sample mean over 30 observations is calculated and reported in Table 4-1. The results at 102.5% RNL are summarized in Table 4-2. The results (the first row of the tables) indicate that the

2. Note that if the demand is identical to the base traffic demand the solution is instantly obtainable since it has been already calculated at the design phase. Thus, we exclude this case from the experiment.

solution procedure typically completes within a few seconds to a few minutes. This implies that the proposed algorithm would be applicable to a network with rapidly changing demand patterns in most cases.

Further speed-up can be attained using the following observations: First of all, observe that only a few commodity path and restoration path variables are generated at the beginning of the solution procedure. The master process generates at most one candidate commodity path variable per commodity and one restoration path variable per arc. If several commodity and restoration path variables are required for each commodity and each arc, which is the case at a higher load, then the solution procedure must iterate between the master process and the sub-process many times before reaching a final solution. If most of the candidate paths are known from the beginning, then unnecessary iterations between the two processes can be avoided. Furthermore, since we have more plausible candidate paths at hand, the objective function value could advance further at each iteration of the sub-process. Thus, we expect to obtain a final solution more quickly.

Candidate paths can be collected by accumulating paths which have been employed over time. The k shortest paths could also be included in the candidate list. The more candidates we have, the

Table 4-1. CPU time for SVPR-MF-LINE 1. 30 random demands at 100% RNL .

Network Model	NJ-LATA		(12,50)		(24,60)		US-WAN	
Test Item	Speed[a]	Loss[b]	Speed	Loss	Speed	Loss	Speed	Loss
MF-LINE	1.04 s	0.6109	1m 13s	0.2454	13.5 s	0.1324	8m 31s	0.1248
MF-LINE w/ RP[c]	0.48 s	0.6109	30.9 s	0.2454	9.71 s	0.1324	3m 58s	0.1248

a. CPU time measured in seconds (s) or minutes (m).
b. An expected lost flow averaged over 30 samples, in terms of a normalized lost flow (NL).
c. The SVPR-MF-LINE with an RP option. (See next page.) K1=10 and K2=30.

Table 4-2. CPU time for SVPR-MF-LINE 2. UF demand at 102.5% RNL

Network Model	NJ-LATA		(12,50)		(24,60)		US-WAN	
Test Item	Speed	Loss	Speed	Loss	Speed	Loss	Speed	Loss
MF-LINE	1.22 s	3.075	2m 23s	3.740	10.6 s	2.643	14m 4s	1.560
MF-LINE w/ RP[a]	1.17 s	3.075	2m 7s	3.740	6.45 s	2.643	5m 11s	1.560

a. K1=10 and K2=30.

fewer iterations are required. However, this could prolong the pricing-out procedure, and thus an excessive generation of the candidates should be avoided. The following approach is applied in our numerical experiments.

- Collect information on the commodity and restoration path variables used in optimal solutions at 100%, 102.5% and 105% RNL.
- Generate all commodity variables employed in the above solutions from the beginning[3].
- For each restoration path, calculate the sum of carried flows in the above solutions. Make an ordered list of the restoration path variables per each arc. List the paths with a higher restoration flow first.
- Append the k shortest restoration paths at the end of the ordered list if they are not included yet. List the paths with a shorter length first.
- Generate K1 restoration path variables per arc from the top of the list at the beginning of the procedure.
- Generate more than K1 but at most K2 restoration path variables per arc if such path variables carry any flow at an optimum.
- Restrict the restoration paths only to the generated ones.

We refer to the SVPR-MF-LINE problem with a restriction on restoration paths as the *SVPR-MF-LINE problem with an RP option*. Due to the restriction, only the dual feasibility conditions related to commodity path variables (Equation (H-4)) must be checked at the master process.

The results are shown in the last row of Table 4-1 and Table 4-2. K1 and K2 are set to 10 and 30, respectively. Restriction on the restoration paths could degrade the attainable survivability measure. Thus, an optimal expected lost flow achieved by each scheme is also listed in the tables for comparative purposes. The amount of lost flow is normalized with respect to the average link capacity. *NL* (normalized loss) is defined as a unit of a normalized lost flow where 100 NL is equal to the average link capacity. The result indicates that we could double or even triple the speed of computation with an RP option. The degradation of attainable survivability is not observed in this experiment, although a small degradation appears if K1 and K2 are set down at 5 and 15, respectively.

3. Based on our experiments, few commodity paths are used at an optimum. Thus, the number of generated path variables is not excessive.

4.3. SVPR-MF-ETE

4.3.1. Problem formulation

The intricate math is shown in Appendix K in order to preserve the flow of the document.

4.3.2. Solution approach

The intricate math is shown in Appendix L in order to preserve the flow of the document.

4.3.3. Validity of the algorithm

The intricate math is shown in Appendix M in order to preserve the flow of the document.

4.3.4. Implementation issues

Due to the increased complexity and size of the problem, the SVPR-MF-ETE problem is expected to require longer computation time than the SVPR-MF-LINE problem. As before, the required CPU time is measured for the four sample networks to see the applicability of the solution procedure. The experiments are conducted with the same conditions as in Section 4.2.4. The results are summarized in Tables 4-3 and 4-4.

The results (the first row of the tables) indicate that except for a small network it takes 20 to 35 minutes to complete the process under normal operating conditions. Thus, end-to-end restoration might not be applicable to a network where a VP reconfiguration procedure must be invoked very frequently. When the network load goes up, the required CPU time grows considerably, suggesting that its applicability is largely restricted at this range of the network load.

As in the case of the SVPR-MF-LINE problem, generating and restricting candidate paths could reduce the computation time. Unlike the previous case, the restoration flow must be obtained for all combinations of commodities and failure events. This requires a significantly large number of restoration path flow variables. Generating many candidate paths, however, could considerably prolong the pricing-out operation. To cope with the problem, the following observations are utilized: First of all, since only a fraction of (π, l) constraints becomes active at a time, the candidate restoration paths can be generated only when the related constraint is active. Secondly, it is possi-

ble to decrease the number of candidate restoration paths corresponding to each (π, l) constraint. In the case of line restoration, arc restoration paths are shared among multiple commodities. On the other hand, (π, l) restoration paths are only prepared for commodity π, and thus not so many paths are needed. Finally, a restriction is also placed on the commodity paths in order to further reduce the computation time.

In the experiments, the candidate paths are collected in the following manner:

- Collect information on the path variables used in the optimal solutions at 100%, 102.5% and 105% RNL. Also, gather information on the path variables which have been generated during the course of the iterations, but stay out of the basis at an optimum.
- For each commodity and restoration path, calculate the sum of carried flow in the above solutions.

Table 4-3. CPU time for SVPR-MF-ETE 1. 30 random demands at 100% RNL

Network Model	NJ-LATA		(12,50)		(24,60)		US-WAN	
Test Item	Speed[a]	Loss[b]	Speed	Loss	Speed	Loss	Speed	Loss
MF-ETE	9.65 s	1.206	20m 28s	1.811	21m 46s	0.5001	34m 27s	0.3076
MF-ETE w/ PRP[c]	8.85 s	1.295	16m 53s	1.939	19m 33s	0.5053	32m 18s	0.3112
MF-LINE	1.04 s	0.6109	1m 13s	0.2454	13.5 s	0.1324	8m 31s	0.1248

a. CPU time measured in seconds (s) or minutes (m).
b. An expected lost flow averaged over 30 samples, in terms of a normalized lost flow (NL).
c. SVPR-MF-ETE scheme with commodity path and restoration path restriction. P1=1, P2=2, RP1=2 and RP2=6. This choice requires the least CPU time based on our experiments.

Table 4-4. CPU time for SVPR-MF-ETE 2. UF demand at 102.5% RNL

Network Model	NJ-LATA		(12,50)		(24,60)		US-WAN	
Test Item	Speed[a]	Loss	Speed	Loss	Speed	Loss	Speed	Loss
MF-ETE	41.3 s	4.812	53m 15s	7.898	2h 36m	5.973	14h 29m	4.002
MF-ETE w/ PRP[b]	16.5 s	4.812	22m 3s	7.898	27m 2s	5.973	1h 41m	4.002
MF-LINE	1.22 s	3.075	2m 23s	3.740	10.6 s	2.643	14m 4s	1.560
MF-LINE w/ RP[c]	1.17 s	3.075	2m 7s	3.740	6.45 s	2.643	5m 11s	1.560

a. CPU time measured in seconds (s), minutes (m) or hour (h).
b. P1=1, P2=2, RP1=2 and RP2=6.
c. K1=10 and K2=30.

- Make an ordered list of the commodity path variables per each commodity. List the paths with a higher traffic flow first. Similarly, make an ordered list of the restoration path variables per each combination of π and l.
- Generate top P1 commodity path variables per commodity from the list. Generate more than P1 but at most P2 variables if such paths have carried some traffic in the optimal solutions.
- When a (π, l) constraint becomes active, generate top RP1 (π, l) restoration path variables from the ordered list. Generate more than RP1 but at most RP2 variables if such paths have carried some restoration traffic in the optimal solutions.

The results are reported in the second rows of Tables 4-3 and 4-4. The above mechanism improves the speed of the solution procedure at 102.5% RNL. However, the required time may not be small enough for a network with rapidly changing demand patterns. Although such a high load would seldom be encountered in a normal situation, it could happen after a network failure. On such an occasion, it would be desirable to perform network-wide restoration at an earlier time. One possible approach is to perform reconfiguration gradually. Since a primal simplex method is employed in the solution procedure, any intermediate solution is feasible. Therefore, a currently available solution could be gradually applied over the course of calculation of the optimal solution.

At 100% RNL, the survivability measure experiences small degradation. Since only a small number of commodity and restoration paths are generated per each commodity and each arc, they could deviate from the paths actually required for the randomly generated demands. Furthermore, no drastic CPU time improvement has been observed at 100% RNL. One possible reason is that the number of constraints is too large to reap the benefit of candidate path generation. It has been empirically observed that the number of simplex iterations grows considerably with the number of constraints and only very slowly with the number of variables [17]. Thus, the benefit of the variable restriction could be cancelled by the explosive number of constraints. Based on the experiment, an almost identical number of pivoting operations is required. As a result, tens of minutes of computational time are still required for most networks. This implies that the applicability of end-to-end restoration is somewhat restricted compared to line restoration.

4.4. SVPR-KSP-LINE

The SVPR-KSP-LINE problem addresses the virtual path routing problem assuming the KSP-

based line restoration as a self-healing protocol. In general, the protocol works as follows: When a link failure happens, the protocol first searches for restoration paths through message-flooding. Every time such a path becomes available, a part of the affected virtual paths are restored using the maximum bandwidth found over the path; that is, the self-healing algorithm first recovers a part of the affected virtual paths by using the entire spare bandwidth available on the shortest restoration route. A subset of the remaining affected virtual paths are restored over the second-shortest route, and further remainders attempt to use the third-shortest route. This procedure is repeated until enough bandwiuth is found to restore all affected VP's or until all possible restoration routes are exhausted. Due to the substantial difference in the restoration process, the problem formulation significantly differs in terms of the restoration flow constraints. The problem turns out to be non-convex, which makes it difficult to find a globally optimal solution. Two heuristic procedures are developed based on two different problem formulations.

4.4.1. Problem formulation

The intricate math is shown in Appendix N in order to preserve the flow of the document.

4.4.2. Solution approach 1. Modified flow deviation (MFD) method

The intricate math is shown in Appendix O in order to preserve the flow of the document.

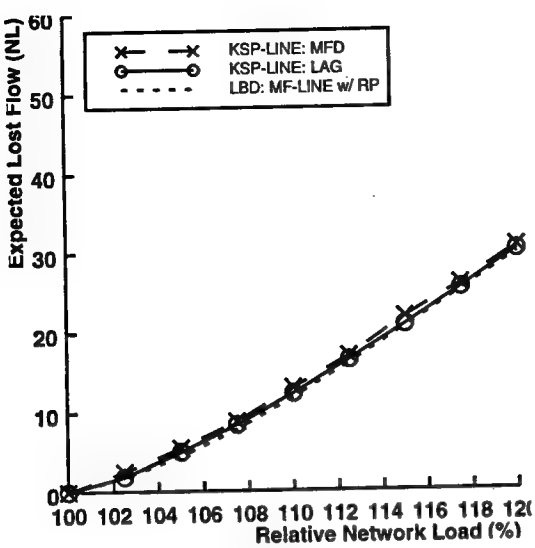
4.4.3. Solution approach 2. Lagrangian relaxation method

The intricate math is shown in Appendix P in order to preserve the flow of the document.

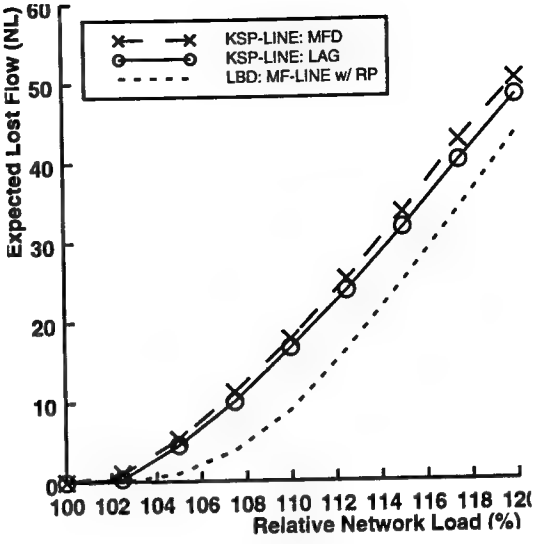
4.4.4. Comparison of two heuristics for the SVPR-KSP-LINE problem

This section evaluates the performance of the two heuristics for the SVPR-KSP-LINE problem proposed in the preceding two sections, the MFD method and the LAG method. Comparative analysis of their attainable expected lost flow is performed based on four sample networks. The uniform traffic (UF) demand and its jointly optimal capacity assignment are utilized in the experiments. The base traffic demand from 100% to 120% RNL is examined. Figure 4-1 depicts the result of the experiments.

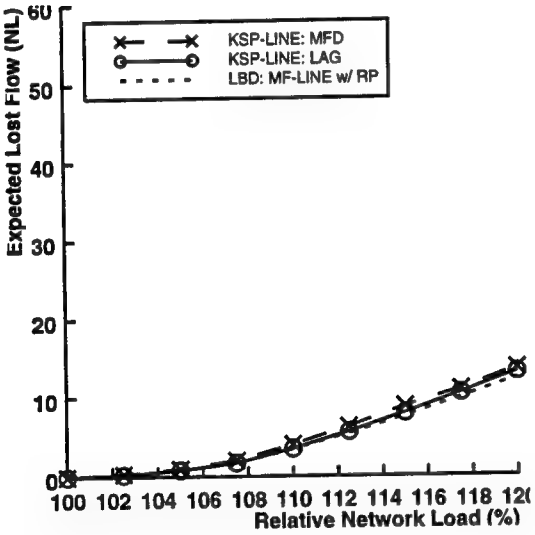
The result shows that the LAG method gives a better solution for all the cases we tested. As



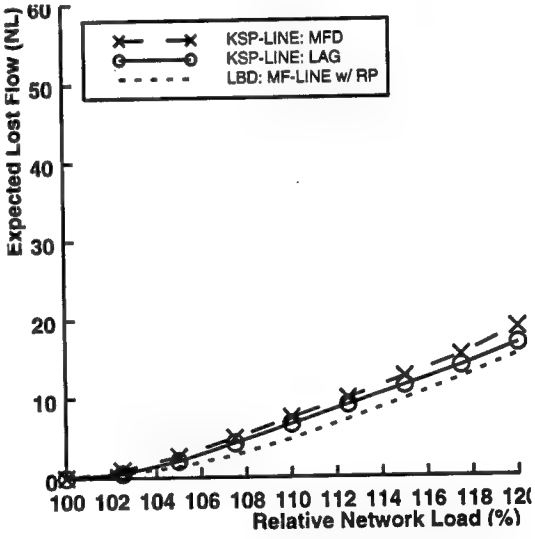
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network



(c) (24,60) Sample Network



(d) US-WAN Sample Network

Figure 4-1. Solution to the SVPR-KSP-LINE problem with two heuristics

In addition to the solutions based on the two heuristics, the lower bound based on the constraint relaxation technique is plotted along the dotted line. The relaxed problem corresponds to the SVPR-MF-LINE problem with an RP option. The minimum value of this relaxed problem produces the absolute lower bound of the solution to the SVPR-KSP-LINE problem.

expected, the LAG method can save more flow as the load grows, up to 2.2 NL at 120% RNL. However, the difference is nominal when the load is light, typically less than 0.5 NL at 102.5% and 105% RNL.

Figure 4-1 also plots the lower bound of the optimal solution along a dotted line. This lower bound is calculated with a constraint relaxation technique (see Appendix S.1), where the KSP constraints (N-10) are completely removed from the problem (SVPR-KSP-LINE2). Then, the resulting minimization problem becomes the SVPR-MF-LINE problem with an RP option. Unlike the Lagrangian subproblem (SVPR-KSP-LINE-LAG), a global minimum is obtainable to this relaxation problem[4]. From the property (S-3) in Appendix S.1, this minimum value is guaranteed to be a lower bound of the SVPR-KSP-LINE problem. However, the bound could be loose since the KSP constraints are not imposed at all. Table 4-5 compares the lower bound and the solution (upper bound) based on the LAG method.

The upper bound is very close to the lower bound for the NJ-LATA and (24,60) networks. The difference is typically less than 0.5 NL, and the gap is mostly less than 5 percent. This indicates that not only is the solution near-optimal, but also the KSP and MF schemes do not make a significant difference in terms of their attainable survivability measure. On the other hand, a wider gap between the upper and lower bounds is observed for the US-WAN network. The gap becomes considerably wider for the (12,50) network; It grows as much as 7.8 NL. This could be due to a large difference between the attainable survivability measures of the KSP and MF schemes. The MF scheme might be able to restore considerably more flow for a large and/or well-connected network. The scheme could distribute a restoration flow more wisely since more candidate restoration paths are available in such networks.

In order to verify this hypothesis, we have performed an intensive random search based on the *SVPR-KSP-LINE-LAG-global* procedure where the KSP constraints are also taken into account. Figure 4-2 shows local minima obtained throughout the *SVPR-KSP-LINE-LAG-global* procedure. One thousand random samples are checked at 105% RNL for the (12,50) network. As discussed in Appendix S, a Lagrangian multiplier largely determines how close the obtained lower bound is to an optimum. The Lagrangian relaxation technique iteratively refines a Lagrangian multiplier to find a tighter lower bound. In the experiment, two distinct Lagrangian multipliers are employed

4. This is because the SVPR-MF-LINE problem with an RP option is a linear programming problem.

which have been encountered at a later stage of the LAG method. The result (b) uses the multiplier at the final iteration of the LAG method, and thus a tighter lower bound is expected.

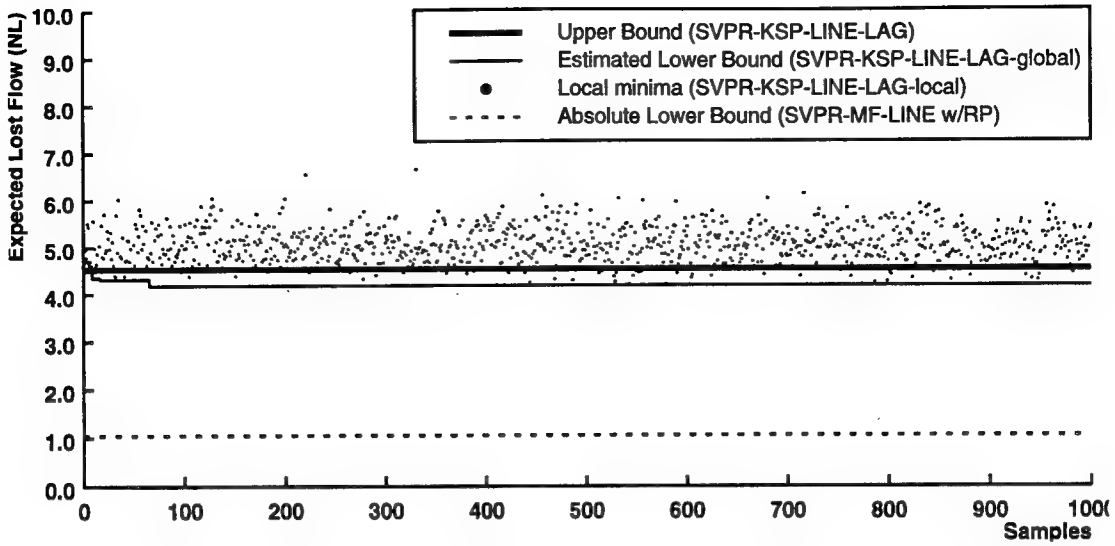
The results suggest the validity of the above hypothesis: The lower bound estimate resides just below the upper bound. The gap is only 0.24 NL or 5.7 percent. On the other hand, the estimate lies 3.24 NL above the absolute lower bound, yielding a more than 13 times wider gap. The same experiment is also conducted at 112.5% RNL, and the result further confirms the hypothesis (Figure 4-3).

Finally, the computation time for the LAG method is significantly longer than that of the MFD method. The LAG method requires as much as tens of minutes to an hour, making it impractical to apply the method to a network with rapidly changing demand patterns. The MFD method, on the

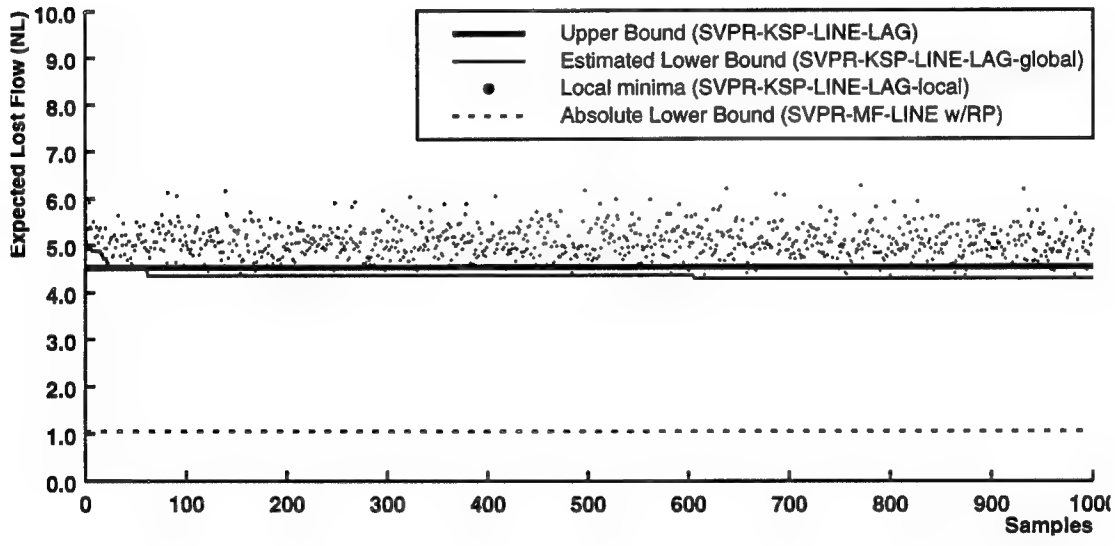
Table 4-5. Upper and lower bounds for the SVPR-KSP-LINE problem

Network Model	Load (RNL)	UBD[a] (NL)	LBD[b] (NL)	Difference[c] (NL)	Tolerance[d] (%)
NJ-LATA	102.5	1.890	1.812	0.079	4.35
	105.0	4.986	4.434	0.551	12.4
	110.0	12.256	11.783	0.473	4.01
	120.0	30.120	29.742	0.378	1.27
(12,50)	102.5	0.364	0.030	0.334	1118
	105.0	4.528	1.046	3.482	333
	110.0	16.707	8.916	7.790	87.4
	120.0	48.182	43.562	4.620	10.6
(24,60)	102.5	0.2555	0.2554	0.0001	0.04
	105.0	0.796	0.796	0.0000	0.00
	110.0	3.495	3.430	0.065	1.90
	120.0	13.086	12.313	0.774	0.88
US-WAN	102.5	0.537	0.537	0.0000	0.00
	105.0	2.085	1.384	0.701	50.6
	110.0	6.685	4.805	1.880	39.1
	120.0	16.748	15.395	1.353	8.79

a. Upper bound. Solution obtained by the LAG method.
b. Lower bound. Optimal solution based on the KSP constraint relaxation.
c. (Upper bound) - (Lower bound)
d. [(Upper bound) - (Lower bound)] / (Lower bound)



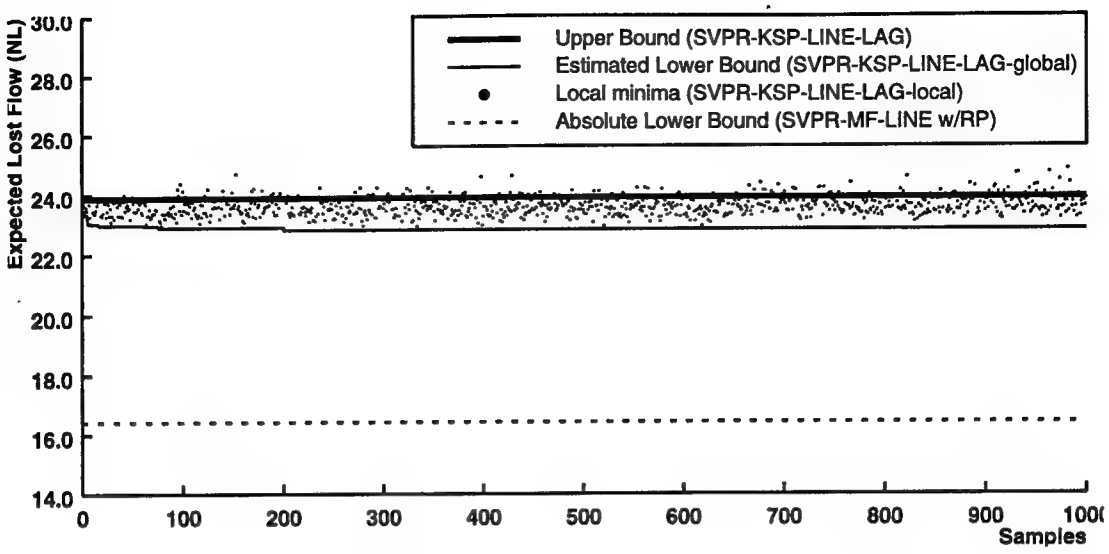
(a)



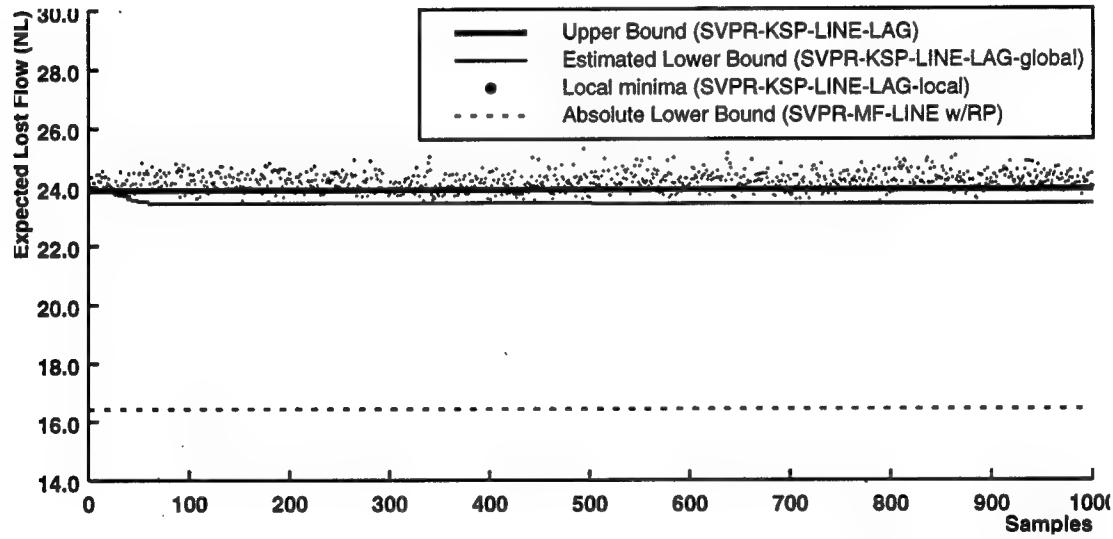
(b)

Figure 4-2. Lower bound estimate for the SVPR-KSP-LINE problem (105% RNL)

The result is for the (12,50) sample network at 105% RNL. The above two results correspond to the two distinct sets of Lagrangian multipliers. Result (b) uses the one encountered at the final stage of the LAG method, yielding a tighter lower bound estimate. The gap between the upper bound and the lower bound estimate is only 0.24 NL or 5.7 percent, while the difference between the estimate and the absolute lower bound is 3.24 NL.



(a)



(b)

Figure 4-3. Lower bound estimate for the SVPR-KSP-LINE problem (112.5% RNL)

The result is for the (12,50) sample network at 112.5% RNL. The above two results correspond to the two distinct sets of Lagrangian multipliers. Result (b) uses the one encountered at the final stage of the LAG method, yielding a tighter lower bound estimate. The gap between the upper bound and the lower bound estimate is only 0.48 NL or 2.0 percent, while the difference between the estimate and the absolute lower bound is 7.0 NL.

other hand, is very quick. Although the MFD method ends up with a higher lost flow, it can usually produce a comparable (near-optimal) solution as long as the load stays around the projected level.

4.5. Evaluations

This section investigates the performance of the proposed survivable VP reconfiguration schemes. First of all, the attainable survivability measure is examined to see how well the proposed optimization procedures work in response to varying traffic demands and at various offered network loads. The effectiveness of dynamic VP reconfiguration is then demonstrated through comparative analysis with a static routing scheme as well as an existing dynamic routing scheme. The three fast restoration schemes are compared in terms of restorability under the same network environment. Finally, the efficiency of the proposed two-step restoration approach is explored.

The eight sample networks which have been scrutinized upon evaluation of the SCFA schemes (Figures 3-4 and 3-5) are employed in the experiments of this section. As discussed in Section I, the physical link capacity assignment is one of the key factors in determining the performance of survivable virtual path assignment as well as fast restoration protocol. Thus, the evaluation of the SVPR schemes on an arbitrarily assigned network resource would not produce meaningful results, although such an approach has often been taken in previous works on self-healing networks. In our experiment, the link capacity assignment is obtained through the SCFA-JOA procedure for a given projected traffic demand (*base traffic demand*). With this design, full service restorability is ensured against any single link failure when the basic traffic demand is offered to the network. The performance of the proposed SVPR procedures is examined at various network loads around the base traffic demand, which would arise in a practical situation. As for the base traffic demand, the uniform traffic demand (UF demand) is utilized in this section unless specified otherwise. The following optimization parameter values are used for the MFD procedure: δ=0.01% and $\varepsilon=10^{-4}$%. The step size, h, is adjusted over the iterations between 10^{-4}% and 0.5% of the average link capacity.

4.5.1. Attainable survivability

This subsection investigates the attainable survivability measure (an expected lost flow) to see how well the proposed SVPR procedures work in various offered loads. In the experiment, the base traffic demands are uniformly increased from 90 to 110 percent of the original base traffic

requirement to examine the performance when the load is light as well as when the traffic demand grows. In the following discussion, this uniformly increased demand is called a *base demand of the load*. A relative network load (RNL)[5] is used as a metric of the network load. The demand of 100% or less RNL represents the case when the network is in a normal operating condition, as discussed before. On the other hand, a load slightly more than 100% RNL demonstrates the case when the traffic demand grows beyond the projected level over months or years, or the case when an unexpected surge of traffic demand arises somewhere in the network. A further heavy load represents the case when the network contains some ill conditions such as multiple network failures or an unexpected overshooting of demand across the network.

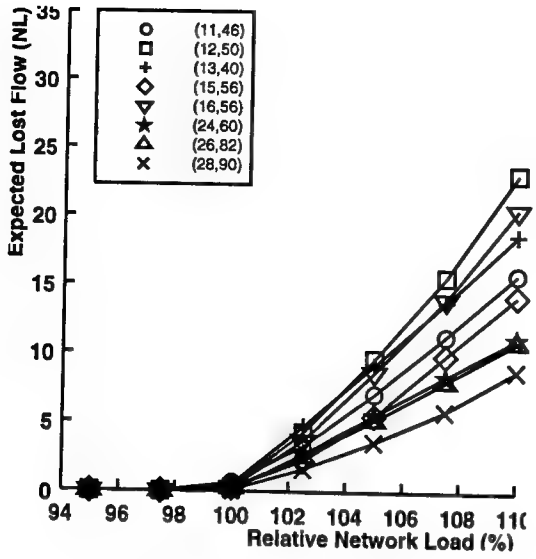
In order to capture the effect of a varying traffic demand, 30 demand patterns are examined at each load: the base demand of the load and 29 additional random traffic demands[6]. The results reported in this subsection are a sample mean over 30 observations. Note that even if the load is 100% or less RNL, the optimal expected lost flow is not necessarily zero unless the demand pattern is identical to that of a base traffic demand.

Figure 4-4 shows the attainable survivability measure for each restoration scheme. The MFD (modified flow deviation) method is utilized for the SVPR-KSP-LINE problem. The result indicates that a lost flow increases very quickly as the offered network load grows beyond the projected demand level. On the other hand, a loss is nominal and typically far below 1 NL with 100% or less RNL, which will be the situation encountered most often in practice. This small loss is due to demand fluctuation. Table 4-6 clarifies the relation between the expected lost flow and the maximum demand fluctuation. Survivability suffers as the demand fluctuates more from the base traffic load, but the degradation is rather insignificant compared with that caused by a demand increase.

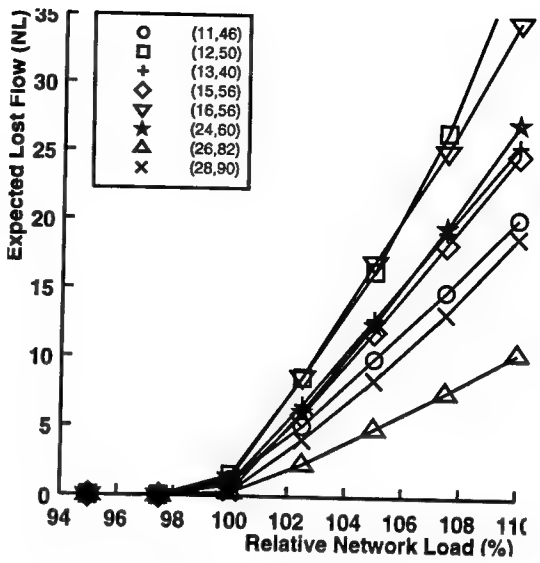
The level of survivability degradation beyond 100% RNL differs according to restoration schemes. The quickest increase on the expected lost flow is observed for the MF-ETE restoration, while the KSP-LINE restoration results in the smallest increase rate. Since an MF-ETE-based network can attain the highest degree of sharing in spare capacity (DSSC) at the link capacity design

5. Recall that a RNL is defined to be 100% if the offered load is same as that projected at the design phase of the network.

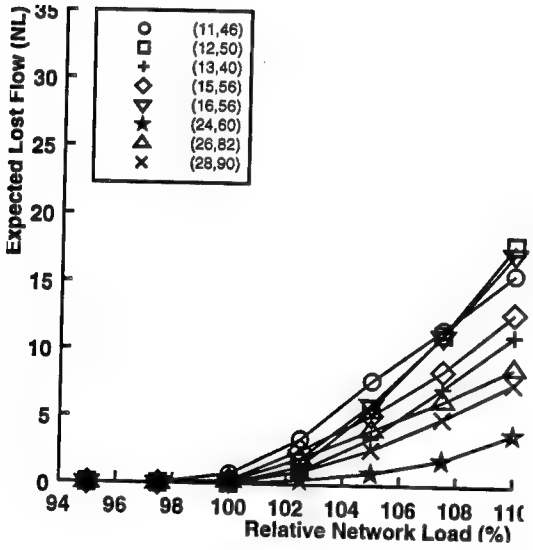
6. As before, the end-to-end traffic requirements of each random input deviate at most 10 percent from those of the base demand of the load. They are uniformly distributed over this range while the total amount of traffic demand is fixed.



(a) MF-LINE scheme



(b) MF-ETE scheme



(c) KSP-LINE scheme

Figure 4-4. Attainable survivability measure for various offered network load

phase, it assigns the least redundant capacity over the network. As a result, the degradation becomes more prominent once the load exceeds the projected level. On the other hand, a KSP-LINE-based network has the most redundant capacity, and thus it is less sensitive to a demand increase. A similar argument holds for the degradation due to demand fluctuation. Table 4-6 shows that an MF-ETE-based network incurs more data loss at the time of a network failure.

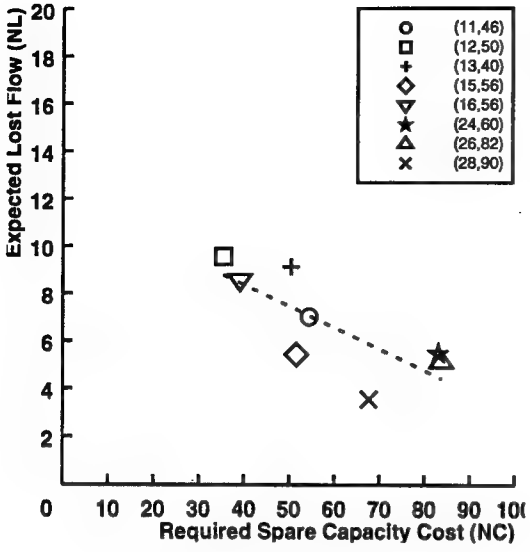
Network models also determine an expected lost flow over 100% RNL. For example, the (12,50) network shows the steepest growth. The (24,60) network has a relatively lower increase rate if the MF-LINE restoration scheme is applied, while the increase is rather quick with the MF-ETE restoration scheme. These phenomena can also be explained by the level of redundant capacity assigned at the design phase: The networks with relatively smaller spare capacity are more vulnerable to network failure at a higher load. Such a relationship between the amount of spare capacity and a lost flow is generally observed at 105% RNL, as illustrated in Figure 4-5. The (24,60) network requires considerably less spare capacity if the MF-ETE restoration is adopted. Therefore, the survivability degradation rates over 100% RNL substantially differ with the two restoration schemes for the (24,60) network.

When the load grows further, a huge loss of data becomes unavoidable upon link failure. At

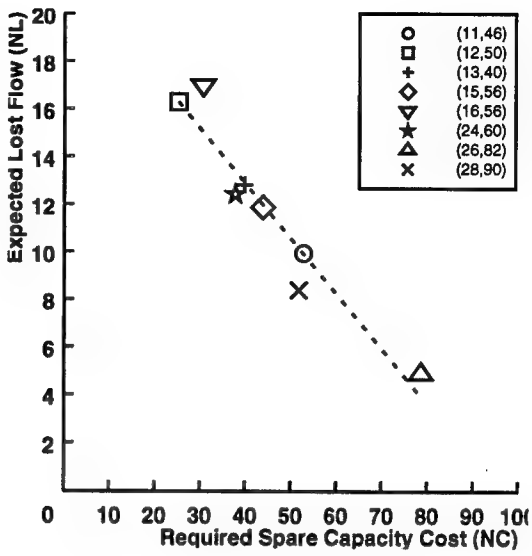
Table 4-6. An expected lost flow versus maximum demand fluctuation

Maximum Demand Fluctuation		5%	10%	15%	20%
MF-ETE	NJ-LATA network model	0.60	1.19	1.56	2.04
	(12,50) network model	0.83	1.81	2.67	3.60
	(24,60) network model	0.26	0.50	0.76	1.02
	US-WAN network model	0.17	0.31	0.49	0.62
MF-LINE	NJ-LATA network model	0.24	0.63	0.96	1.51
	(12,50) network model	0.06	0.25	0.53	0.93
	(24,60) network model	0.05	0.14	0.20	0.28
	US-WAN network model	0.05	0.13	0.20	0.28
KSP-LINE	NJ-LATA network model	0.29	0.70	0.92	1.48
	(12,50) network model	0.02	0.06	0.14	0.26
	(24,60) network model	0.00	0.00	0.01	0.02
	US-WAN network model	0.03	0.06	0.10	0.14

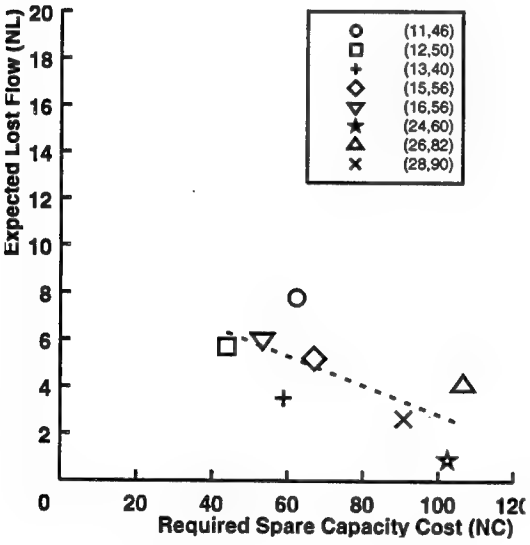
Thirty random demands are used in the experiment. A sample mean of 30 observations is reported. An average expected lost flow is expressed in NL (normalized loss)



(a) MF-LINE scheme



(b) MF-ETE scheme



(c) KSP-LINE scheme

Figure 4-5. A lost flow at 105% RNL versus normalized spare capacity cost

The UF demand pattern is employed as a base traffic load. The normalized lost flow is a sample mean over 30 samples at 105% RNL: one base demand plus 29 randomly fluctuated demands. End-to-end traffic requirements of each random demand deviate at most 10 percent around the base demand and are uniformly distributed over this range.

110% RNL, the expected lost flow becomes as much as 40 NL. As discussed above, though, this is not in the normal operating region. The load may increase up to this range, but this must happen very infrequently, possibly due to an occasional surge of traffic or multiple network failures. However, a higher level of network survivability cannot be assured if the load stays in this region due to growing traffic demand. Therefore, the FN manager must be triggered to install additional network resources before the traffic increases to this level. As mentioned above, a lost flow develops very quickly for the networks with a higher DSSC. This suggests that the design process for an additional capacity installation should be planned at an earlier stage of the demand growth in such a network in order to maintain a desired survivability measure.

4.5.2. Effectiveness of dynamic VP reconfiguration

4.5.2.1. Comparison with a static routing scheme

We propose a dynamic virtual path routing scheme in order to attain better survivability in response to demand dynamics. Previous works in self-healing networks assume a predetermined flow assignment. This section manifests the effectiveness of the proposed dynamic VP reconfiguration scheme through comparative analysis with a static routing scheme.

In contrast to a dynamic routing scheme, a static routing scheme utilizes predetermined routes and bandwidth allocation strategy. Control simplicity is a major advantage on a static routing scheme, but it comes at the expense of survivability and network utilization. The static routing scheme employed in this experiment exploits the information on the flow assignment obtained in the network design. Recall that the joint optimization procedure of the SCFA outputs not only a capacity allocation but also an optimal virtual path routing and bandwidth assignment for a projected traffic demand. In the static routing scheme, the traffic demand of each commodity is sent over the same routes as used in the optimal VP assignment. If multiple routes are employed for a commodity, its demand is distributed over the routes at the same ratio as obtained in the design phase. Full restorability is assured in this assignment if a projected traffic demand is offered to the network. In the following discussion, this static routing scheme is referred to as the SSR (survivable static routing) scheme, while our proposed routing method is called SDR (survivable dynamic routing) scheme[7].

Thirty random demands deviating around the projected traffic demand (100% RNL) are exam-

ined in the experiment. Each random input has at most 10 or 20% demand deviation from the base traffic demand. The average expected lost flow due to the SSR and SDR schemes is reported in Table 4-7. The table also lists the average survivability gain. It is defined by $(\hat{L}_{SSR} - \hat{L}_{SDR}) / \hat{L}_{SSR}$ where $\hat{L}_{SSR}$ and $\hat{L}_{SDR}$ are the average expected lost flow attained by the SSR and SDR schemes, respectively[8]. The MFD method is assumed for the dynamic routing strategy for KSP-LINE-based networks. In some cases, the traffic demand cannot even be accommodated with the static routing scheme due to capacity constraints. Such a flow is counted lost in the above calculation.

A significant improvement on the survivability measure is attained by dynamic routing. More than 50 percent survivability gain is observed in most cases. Out of the three types of networks, an MF-ETE-based network turns out to have a relatively low survivability gain. As discussed in Section 4.5.1, a lost flow in such a network could be higher than the other cases even with dynamic routing due to a smaller amount of spare bandwidth installation. Furthermore, the efficiency of end-to-end restoration could make restorability less sensitive to flow assignment. As a result, a low survivability gain follows, although as much as 5 NL lost flow could be saved.

On the other hand, the highest survivability gain can mostly be attained with an MF-LINE-based network. This phenomena can be explained by completely opposite reasoning. A larger amount of spare capacity enables recovering more traffic even with demand fluctuation for dynamic routing. On the other hand, the line restoration scheme is not so efficient as the end-to-end restoration scheme in terms of restoration capability. Therefore, restorability becomes more sensitive to flow assignment. As a result, it could save a lost flow as much as 9 NL on average.

Lost flow is not so significant for a KSP-LINE-based network for both routing strategies. Relatively more spare capacity is installed in a KSP-LINE-based network than any other type of networks. This is not only because KSP restoration requires more redundant bandwidth for data recovery, but also because the capacity assignment for such networks has been calculated through heuristics. The result indicates that this additional capacity gives enough room to reduce a lost flow even with static routing. With dynamic routing, a lost flow becomes nominal for a KSP-

7. Note that the SSR is based on the output of the survivable network design. Therefore, the scheme is expected to attain a higher survivability measure than a routing scheme without any consideration of network survivability, such as the dynamic routing scheme discussed in the next section.

8. The hat sign (^) means a sample mean of the 30 observations.

Table 4-7. Comparison between SSR and SDR schemes 1. 100 RNL

Network Model		NJ-LATA		(12,50)		(24,60)		US-WAN	
Maximum deviation		10%	20%	10%	20%	10%	20%	10%	20%
MF-ETE	Average loss by SSR[a] (NL)	2.17	4.45	3.50	8.70	0.75	1.57	0.38	0.79
	Average loss by SDR (NL)	1.19	2.04	1.81	3.60	0.50	1.02	0.31	0.62
	Survivability gain (%)	45.0	54.1	48.2	58.6	33.4	34.9	19.9	21.7
	Spare capacity (NC)[b]	52.86		25.31		37.80		51.86	
MF-LINE	Average loss by SSR (NL)	1.93	4.30	2.81	4.05	2.55	2.69	9.20	9.51
	Average loss by SDR (NL)	0.63	1.51	0.25	0.93	0.14	0.28	0.13	0.28
	Survivability gain (%)	67.3	64.9	91.0	77.1	94.6	89.6	98.6	97.0
	Spare capacity (NC)	54.45		35.52		83.07		67.65	
KSP-LINE	Average loss by SSR (NL)	1.44	3.24	0.34	0.80	0.052	0.16	0.13	0.28
	Average loss by SDR (NL)	0.70	1.48	0.058	0.26	0.0015	0.023	0.062	0.14
	Survivability gain (%)	51.6	54.5	82.7	66.7	97.2	85.3	50.5	50.4
	Spare capacity (NC)	62.41		43.99		102.58		90.87	

a. The row entries of 'average loss' list the expected lost flow averaged over 30 random samples.
b. Spare capacity installed in the network in terms of normalized capacity (NC).

Table 4-8. Comparison between SSR and SDR schemes 2. 105 RNL

Network Model		NJ-LATA		(12,50)		(24,60)		US-WAN	
Maximum deviation		10%	20%	10%	20%	10%	20%	10%	20%
MF-ETE	Average loss by SSR (NL)	15.79	18.77	28.78	35.59	17.37	18.23	11.49	12.42
	Average loss by SDR (NL)	9.92	10.65	17.28	18.50	12.47	12.99	8.38	8.63
	Survivability gain (%)	37.2	43.3	39.9	48.0	28.2	28.8	27.0	30.5
	Spare capacity (NC)	52.86		25.31		37.80		51.86	
MF-LINE	Average loss by SSR (NL)	11.00	12.87	13.40	15.60	7.25	7.33	16.75	17.11
	Average loss by SDR (NL)	7.09	7.70	9.66	10.28	5.47	5.46	3.57	3.64
	Survivability gain (%)	35.6	40.1	27.9	34.1	24.6	25.4	78.7	78.7
	Spare capacity (NC)	54.45		35.52		83.07		67.65	
KSP-LINE	Average loss by SSR (NL)	9.20	10.65	7.25	7.90	2.91	2.98	5.00	5.11
	Average loss by SDR (NL)	7.33	8.29	5.61	6.22	0.91	0.97	2.63	2.68
	Survivability gain (%)	20.4	22.2	22.6	21.3	68.6	67.3	47.4	47.7
	Spare capacity (NC)	62.41		43.99		102.58		90.87	

LINE-based network.

We have further carried out the same experiments at 105% RNL to examine the effectiveness of the SDR scheme at a higher load. The results are summarized in Table 4-7. As expected, a larger amount of lost flow can be avoided by the SDR scheme than the case at 100% RNL. Due to a higher load offered to the network, a dynamic flow adjustment is more effective to reduce an expected lost flow. However, the average survivability gain decreases because more lost flow is inevitable even with the SDR scheme.

4.5.2.2. Comparison with an existing dynamic routing scheme

This section explores the effectiveness of the proposed routing scheme through a comparison with an existing dynamic routing strategy. Recall that no existing routing scheme has aimed at promoting the restorability of a self-healing protocol. The objective of this comparative analysis is to examine the improvement of attainable restorability by the introduction of the survivable control at the VP layer. Furthermore, we attempt to clarify how the SDR scheme can enhance the survivability measure by analyzing the difference in the obtained flow assignments.

Out of possible candidates for a comparison, we have chosen the traditional dynamic routing strategy in packet switching networks where average network delay is minimized. A flow deviation method can be employed to solve this minimization problem [22]. This scheme is referred to as the DR (dynamic routing) scheme in the following discussion. A close relation between average delay and buffer overflow probability suggests that the DR scheme effectively reduces the cell level QOS parameter (cell loss probability) to a near-optimal level by minimizing the average delay. In fact, this approximation has been used before in the context of dynamic ATM path reconfiguration [62]. Furthermore, the DR scheme could avoid more lost flow than other traditional routing strategies. The scheme tends to balance the traffic load over the network, and thus spare bandwidth is allocated evenly over the network. Therefore, the results obtained through comparative analysis between the DR and SDR schemes would show performance improvement over one of the best possible routing schemes proposed so far.

In the experiments, the base traffic demands are uniformly increased from 90 to 110% RNL. Thirty random demand patterns are utilized for each load. The random inputs deviate at most by 10

or 20 percent around the base demand and are uniformly distributed over the range. The KSP-LINE-based network is examined in this study. The MF-based scheme requires an additional pre-planning process to find a restoration flow assignment, but such a provision is not provided in any existing dynamic routing strategy.

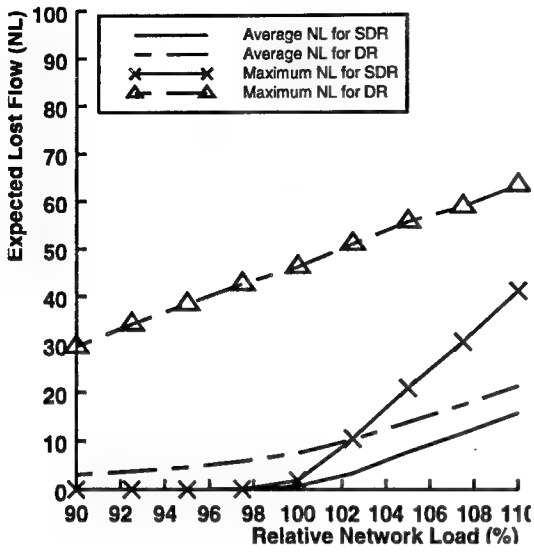
Figures 4-6, 4-7 and 4-8 show the results of the experiments on the four sample networks. They are based on the uniform traffic (UF) demand pattern, but the results over other base demand patterns (WT and RD demand patterns) turn out to exhibit a similar trend for each network model. This indicates that the results have no significant dependency on the base traffic demand as long as the link capacity is properly designed according to the projected demand. In Figure 4-6, an expected lost flow as well as a maximum lost flow per single-link failure, $\max_{l \in E} \hat{L}_l$, are plotted against the offered load. Figure 4-7 illustrates the improvement of survivability measure attained by the SDR scheme over that of the DR scheme. An improvement on an expected lost flow as well as its maximum and minimum per single-link failure are shown in the figure. Finally, Figure 4-8 depicts the average survivability gain, $(\hat{L}_{DR} - \hat{L}_{SDR}) / \hat{L}_{DR}$, where $\hat{L}_{SDR}$ and $\hat{L}_{DR}$ are the average expected lost flow attained by the SDR and DR schemes, respectively. The results at 100 and 105 RNL are also listed in Table 4-9.

The maximum improvement of a lost flow due to a single link failure amounts to 5 to 45 NL at 100% RNL, while the average is at most 7 NL. The average gain is not significant since with some links a flow can be restored in either flow assignment scheme from their failure. For example, the

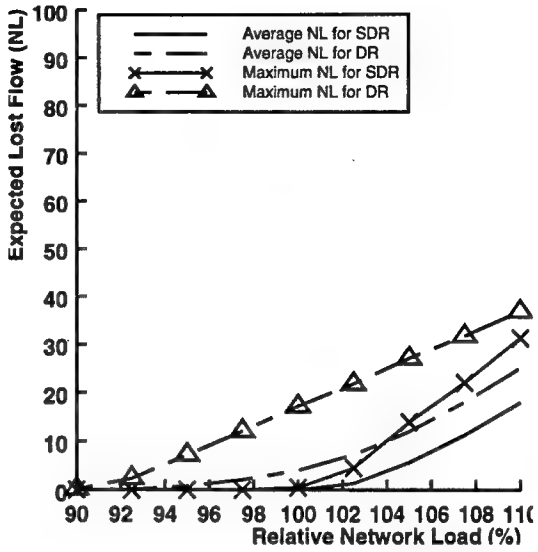
Table 4-9. Comparison between DR and SDR schemes[a]

Load	Network Model	NJ-LATA		(12,50)		(24,60)		US-WAN	
	Maximum deviation	10%	20%	10%	20%	10%	20%	10%	20%
100 RNL	Average loss by DR[b] (NL)	7.54	7.68	4.06	4.12	0.94	1.09	4.27	4.21
	Average loss by SDR (NL)	0.70	1.48	0.058	0.26	0.0015	0.023	0.062	0.14
	Survivability gain (%)	90.7	80.7	98.6	93.7	99.8	97.9	98.5	96.7
	Average loss by SSR (NL)	1.44	3.24	0.34	0.80	0.052	0.16	0.13	0.28
105 RNL	Average loss by DR (NL)	13.85	13.53	11.99	11.88	2.54	2.63	7.10	6.86
	Average loss by SDR (NL)	7.33	8.29	5.61	6.22	0.91	0.97	2.63	2.68
	Survivability gain (%)	47.1	38.7	53.2	47.6	64.2	63.1	63.0	60.9
	Average loss by SSR (NL)	9.20	10.65	7.25	7.90	2.91	2.98	5.00	5.11

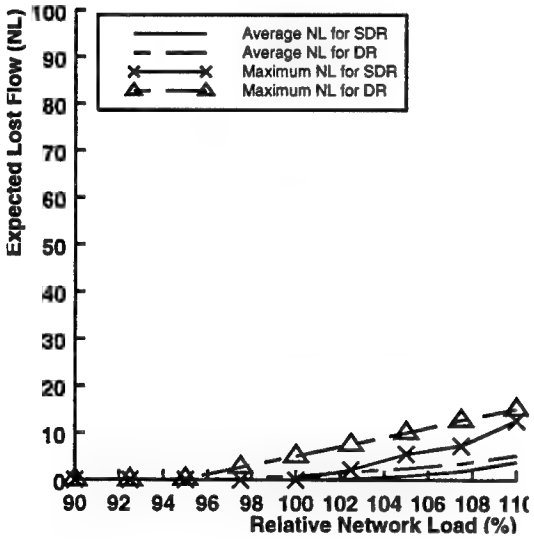
a. The KSP-LINE restoration scheme is employed.
b. The row entries of 'average loss' list the expected lost flow averaged over 30 random samples.



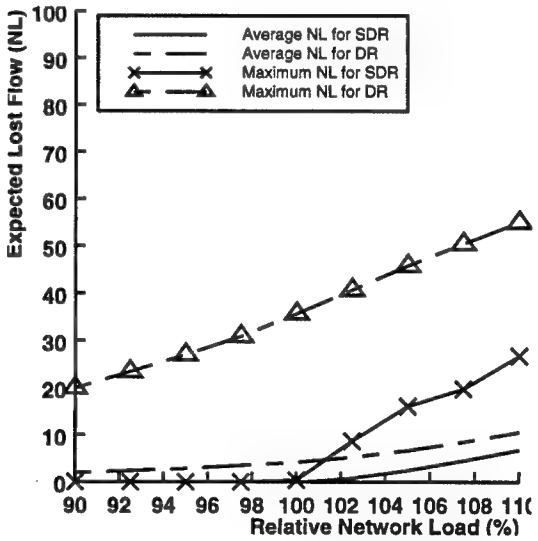
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network



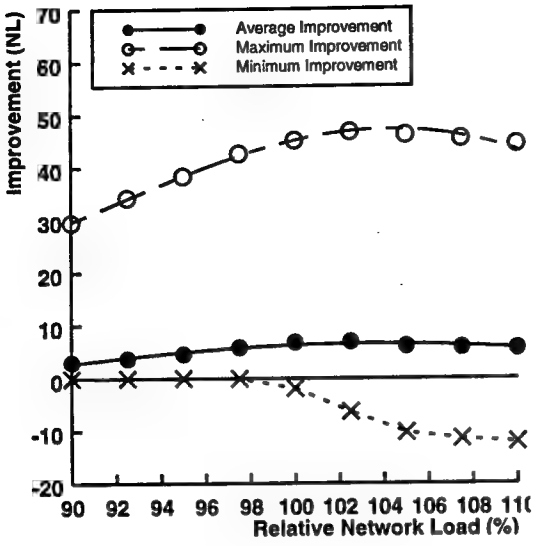
(c) (24,60) Sample Network



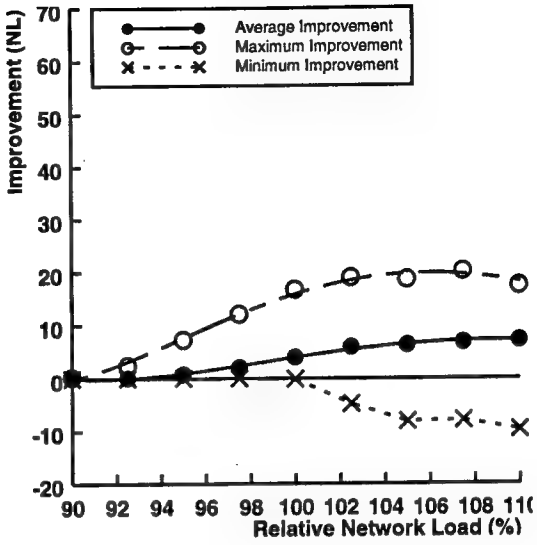
(d) US-WAN Sample Network

Figure 4-6. Comparison of a lost flow in the SDR and DR schemes

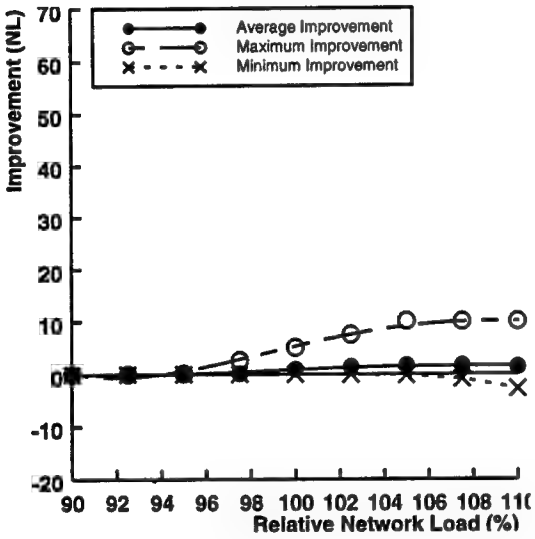
Thirty random demands, which deviate at most 10 percent from the base demand, are employed at each load



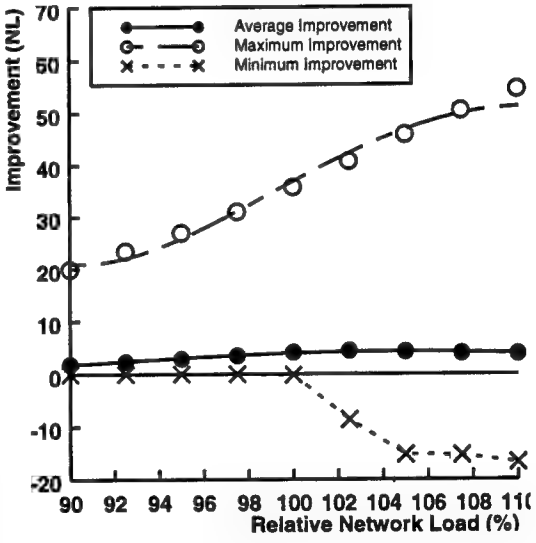
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network



(c) (24,60) Sample Network



(d) US-WAN Sample Network

Figure 4-7. Improvement of survivability measure by SDR over DR

Thirty random demands, which deviate at most 10 percent from the base demand, are employed at each load

links 1-5, 1-8 and 5-8 in the NJ-LATA network have many disjoint restoration routes (see Figure 3-4), and it is observed that a lost flow due to a failure of these links is zero in either approach. On the other hand, significant improvement is observed in a failure of the link adjacent to a node with degree 2, such as links 5-7 and 7-8 in the NJ-LATA network model. If such a link fails, all active flow must be rerouted over its adjacent link. Thus, relatively more spare capacity must be allocated over these links in order to achieve a high restorability. The DR scheme fails to assign an extra spare capacity to such links because it tries to balance the load uniformly over the network. The SDR scheme, on the other hand, can allocate the traffic in such a way that a flow would not be lost in such links. In other words, our algorithm can detect the links vulnerable to a failure under the current traffic demand patterns and avoid allocating excessive flow on these links in order to promote restorability over the network.

The attainable survivability measure and improvement also differ according to the network topology. Out of the four sample networks, the (24,60) network produces the least lost flow and attains the least improvement by the SDR scheme. On the other hand, a relatively large average lost flow is observed in the (12,50) and NJ-LATA networks. The amount of installed spare capacity again explains the difference of an average lost flow: A network with less spare bandwidth results in a larger expected lost flow at a higher load. However, the maximum improvement per failure is not so large in the (12,50) network as in the NJ-LATA and US-WAN networks, although the least spare capacity is installed in the (12,50) network. This is because the (12,50) network does not include a node with degree 2. As mentioned above, the links adjacent to such a node make the network vulnerable to a more damaging failure. The DR scheme fails to avoid allocating an excessive flow over the links, and thus a failure of those links significantly contributes to a large improvement by the SDR scheme.

When the load is not heavy, the SDR scheme works very well to prevent data loss. The maximum lost flow is almost zero with 100% or less RNL, implying that virtually no flow is lost with any single-link failure at this range of operation. On the other hand, a flow as much as 40 NL can be lost by the DR scheme. As discussed before, the region with 100% or less RNL is the one encountered in normal operating conditions, and our approach almost achieves 100% survivability gain over this range. Even if the demand occasionally increases slightly beyond the projected level, the SDR scheme can prevent most of the lost flow. For example, about 80% of the lost flow can be saved when the demand increases by 2.5%.

When the load increases further, the survivability gain becomes small, about 50% with the load at 105% RNL and around 30% at 110% RNL. At this region of the load, the minimum improvement becomes negative. Namely, the flow due to the DR scheme produces a smaller lost flow than that due to the SDR scheme for some link failures. This is possible since the SDR procedure optimizes the expected lost flow over all possible single link failures. Lost flow is inevitable against any link failure when the load gets heavy. Thus, it becomes almost impossible to find a flow which is more restorable than others against any link failure. On average, our approach still gives some improvement over the DR scheme. As discussed before, 110% RNL is not in the normal operating region. If the load stays in this region due to growing traffic demand, data loss becomes unavoidable with any routing strategy. A failure at the most vulnerable link entails a huge service interruption. For example, 41 NL may be lost for some link failures in the NJ-LATA network. In other words, the load is too heavy to prevent large loss with the available network resources. Thus, additional resources must be installed before the load increases to this level.

4.5.3. Restorability of three VP restoration schemes

This section reports the results of comparative analysis on the restorability of the three fast restoration schemes: MF-LINE, MF-ETE and KSP-LINE. Attainable survivability measures are compared under the same network environment: the same capacity placement and offered load. The capacity assignment is calculated by the SCFA-MF-ETE scheme for a given projected traffic demand[9]. A flow assignment is optimized using the respective VP reconfiguration procedures, and

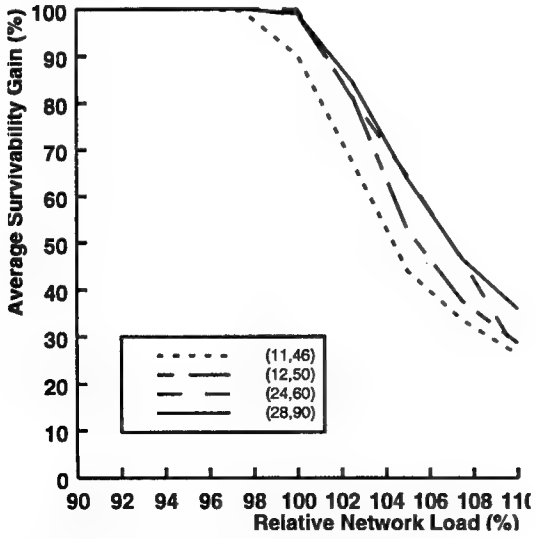


Figure 4-8. Survivability gain of SDR over DR

9. The UF demand pattern is used in this experiment.

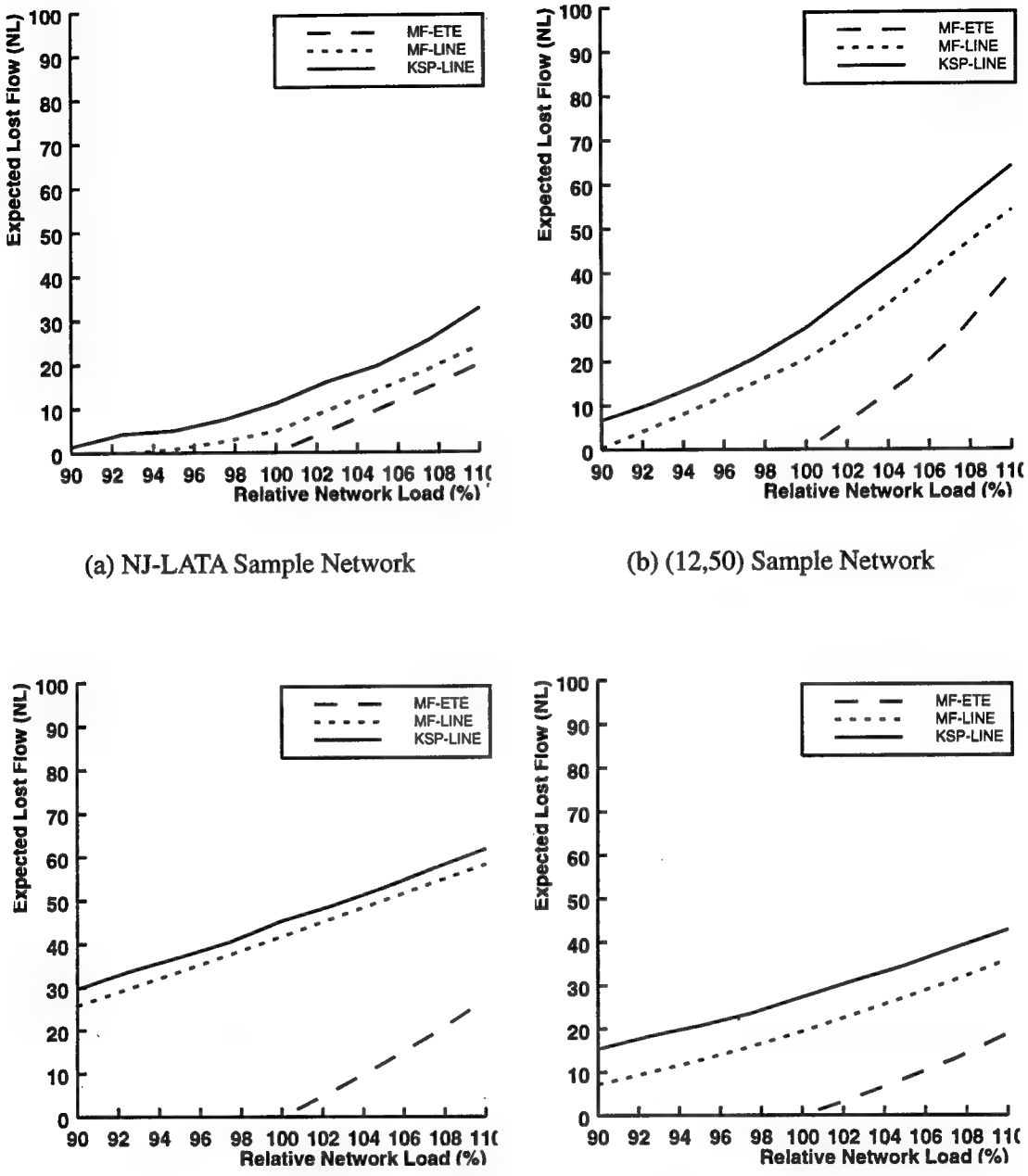
the restoration is performed by the corresponding restoration schemes. Figure 4-9 summarizes the results of this study for four sample networks. The load is increased from 90% to 110% RNL, and only a base demand is employed at each load.

As expected, the MF-ETE scheme attains the highest restorability, while the KSP-LINE scheme suffers from the largest loss of data. However, the relative performance of each scheme significantly differs according to network topologies. For example, the MF-ETE scheme has a great advantage over the other two schemes in the (24,60) network, while the difference between the MF-LINE and the KSP-LINE schemes is insignificant. At 100% RNL, the MF-ETE scheme could save about 42 NL and 45 NL lost flow over the MF-LINE and KSP-LINE schemes, respectively. The NJ-LATA network, on the other hand, does not give a substantial benefit on the MF-ETE scheme: A lost flow conserved over the MF-LINE is only about 4 NL at 100% RNL.

Regularity and connectivity of the network, which have been widely discussed in the previous section, again account for these phenomena. Since the (24,60) network is well-balanced, the end-to-end restoration scheme can distribute restoration flow evenly over the network. In addition, the sparseness and low node connectivity of the network make the line restoration scheme more inefficient. Since all restoration flow must be sent between two nodes adjacent to a failed link, a restorable amount of flow is restricted to the total available spare bandwidth leaving the nodes. A network with a lower node degree is severely affected by this bandwidth restriction, imposing a significant penalty on the restorability of the line restoration scheme. Furthermore, such a network is more susceptible to the effect of backhauling and looping, as discussed in the previous section. In the end, the MF-ETE scheme achieves a substantial improvement for the (24,60) network not only due to its effectiveness but also due to the inefficiency of the line restoration scheme for the network. On the other hand, since the NJ-LATA network is well-connected, the locality of restoration flow would not place a large penalty on the line restoration scheme. Irregularity of the network topology might diminish the benefit of the end-to-end restoration scheme. Therefore, the MF-ETE scheme is less attractive for the NJ-LATA sample network.

4.5.4. The effect of two-step restoration

Two-step restoration aims at not only performing a quick recovery from a network failure but also obtaining an optimal flow assignment against a possible subsequent network failure. Network-wide restoration (NWR) is performed to realize an optimal VP assignment after temporal



(a) NJ-LATA Sample Network

(b) (12,50) Sample Network

(c) (24,60) Sample Network

(d) US-WAN Sample Network

Figure 4-9. Restorability of the three SVPR schemes

service interruption is recovered by fast restoration mechanisms. This section investigates the effect of the two-step restoration scheme through a comparison of attainable survivability with and without the NWR after a failure. An expected lost flow due to two successive link failures is used to measure the performance. All possible combinations of link failures are considered, and they are assumed to be equally likely to occur. Formally, an expected lost flow due to two successive link failures, $L^2(\underline{\mathbf{f}})$, is defined as follows:

$$L^2(\underline{\mathbf{f}}) = \frac{1}{|E|}\sum_{l \in E} L_l^2(\underline{\mathbf{f}})$$

$$L_l^2(\underline{\mathbf{f}}) = L_l^1(\underline{\mathbf{f}}) + \frac{1}{|E|-1}\sum_{e \in E, e \neq l} L_e^{2\circ}(\hat{\underline{\mathbf{f}}}_l)$$

where two complete link cuts are assumed to happen at links l and e consecutively. $L_l^1(\underline{\mathbf{f}})$ is a lost flow upon restoration of the first failure at link l, where $\underline{\mathbf{f}}$ is a flow assignment before the failure. $L_e^{2\circ}(\hat{\underline{\mathbf{f}}}_l)$ is a lost flow due to the second failure at link e which happens under a new topology without link l and under a new flow assignment, $\hat{\underline{\mathbf{f}}}_l$. $\hat{\underline{\mathbf{f}}}_l$ is a flow obtained after a failure of link l. Note that $\hat{\underline{\mathbf{f}}}_l$ takes a different value depending on the availability of the NWR function. If some amount of flow cannot be recovered upon restoration of the first failure, a part of the VP demand cannot be satisfied after the failure. In such a case, a part of the requested VP bandwidth must be rejected. If the NWR is applied, however, the flow assignment $\hat{\underline{\mathbf{f}}}_l$ largely depends on how the demand is discarded. In the calculation of $\hat{\underline{\mathbf{f}}}_l$, we assume that this demand rejection takes place uniformly over the affected VP's. Finally, $L_l^2(\underline{\mathbf{f}})$ gives an expected lost flow due to two successive link failures, assuming the first failure occurs at link l.

The relative network load from 80 to 105 percent is explored in the experiment. Only the base demand is used at each load. The initial flow before the first failure, $\underline{\mathbf{f}}$, is assumed to be computed by the proposed VP assignment procedures. The three restoration strategies are considered, and the MFD method is employed in the experiments on KSP-LINE-based networks. After fast restoration from a failure, some of the restored VP's might be routed back to the link previously encountered in the case of line restoration (backhauling). For example, a VP with a route 2-3-4 in the US-WAN network (see Figure 3-4) may be rerouted over path 2-3-2-7-8-4 upon a link 3-4 failure, and the path goes back and forth over the link 2-3. For KSP-LINE-based networks, the NWR procedure removes these unnecessary loops before triggering the optimization process. Then, the procedure uses the resulting flow assignment as its initial solution. This preprocessing is desirable since such loops may not be removed completely by the algorithm. Since the algorithm usually finds a new

flow along the line between the current flow and a shortest route flow, some amount of bandwidth tends to remain allocated over the loops. In addition, it is beneficial to get rid of these obviously inefficient routes beforehand because the iterations needed for their removal can be eliminated. This preprocessing is unnecessary for MF-LINE-based networks since the solution procedure does not use the current solution as a starting point.

The results are illustrated in Figures 4-10 to 4-18. An average normalized lost flow, $L^2(\underline{f})$, is depicted in Figures 4-10, 4-13 and 4-16 for both cases with and without the NWR function. Figures 4-11, 4-14 and 4-17 demonstrate the improvement of survivability measure with the NWR function. The maximum improvement on L_l^2 $(l \in E)$ is depicted in addition to the average improvement on L^2. Finally, Figures 4-12, 4-15 and 4-18 show the average survivability gain due to the NWR, given by $(L_{WO} - L_W)/L_{WO}$. L_W and L_{WO} denote the lost flow, L^2, with and without the NWR, respectively. The figures also plot the maximum survivability gain at each load, which is defined by $\max_{l \in E}\{(L_{lWO}^2 - L_{lW}^2)/L_{lWO}^2\}$. L_{lW}^2 and L_{lWO}^2 give the expected lost flow, L_l^2, due to a failure of link l and another subsequent link failure with and without the NWR, respectively.

In the case of line restoration-based networks, on average 10 to 30 NL lost flow can be avoided with the NWR at 100% RNL, 21 to 68 NL at most (Figures 4-11 and 4-14). The maximum survivability gain is 100% at a lower load (Figures 4-12 and 4-15). This implies that the NWR can completely restore a flow against a certain scenario of successive link failures. For example, suppose the link 5-8 fails in the KSP-LINE-based NJ-LATA network and the load is below 90%. If the NWR function is used, a flow can be restored from any successive failure, while on average 6 NL flow is lost at the time of the second failure without the NWR function. Note that the average lost flow due to two successive failures takes a positive value even with 80% relative load. More capacity is required to realize full restorability against multiple link failures.

The average and maximum improvement increases as the load grows to the projected level. Due to the reduced amount of spare capacity, a flow must be assigned more efficiently at a higher load in order to restore more bandwidth against successive outages. This indicates that the NWR function is an effective way to promote network survivability at the presence of a network failure. Further growth on the network load, however, causes the improvement to saturate and decrease. At this saturation level, data loss becomes inevitable even with the NWR function. Such saturation

occurs at a lower load for the (12,50) network due to a smaller spare capacity installation.

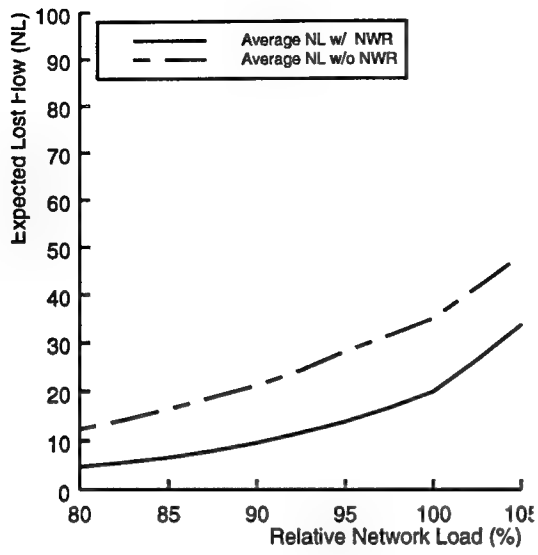
In the case of end-to-end restoration-based networks, the improvement of the survivability is significantly smaller (Figure 4-17). On average, 6 to 9 NL lost flow can be avoided with the NWR at 100% RNL. Survivability gain is also considerably smaller than that for line restoration-based networks (Figure 4-18). End-to-end restoration is immune to backhauling and can exploit the spare bandwidth more effectively than line restoration, even without the NWR function. The smaller improvement is due to this efficiency of the restoration mechanism. Furthermore, less spare capacity is installed in ETE-based networks, making them more susceptible to multiple failures. Thus, a relatively larger lost flow is incurred even with NWR. The expected lost flow due to two successive failures increases rather quickly at a higher load, as observed in Figure 4-16. Because of a smaller improvement and a larger amount of data loss, the scheme can only attain a smaller survivability gain than the other schemes.

4.6. Summary

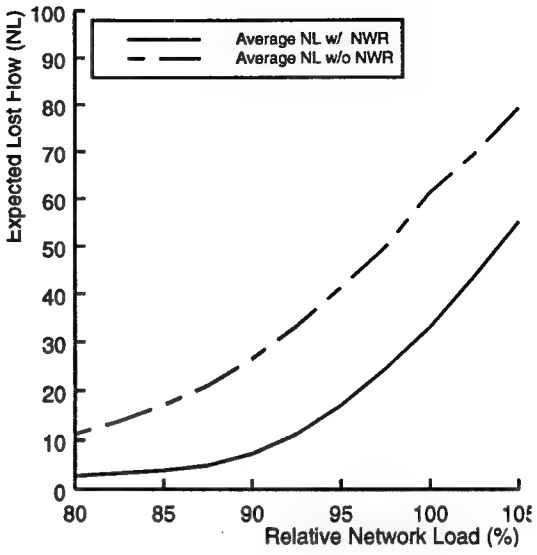
This section presents survivable virtual path routing problems for self-healing ATM networks based on three fast restoration protocols. A similar approach to the SCFA problem can be applied to the MF-based networks, while two novel algorithms are developed for the KSP-based networks. The modified flow deviation method is developed to solve a non-smooth multicommodity flow problem. A near-optimal solution can be found instantly while avoiding kinks by adjusting an optimization parameter. A Lagrangian relaxation-based approach further advances the quality of the solution although it requires more computation time. This approach also helps to obtain a lower bound estimate of the optimum.

Numerical experiments reveal that the proposed algorithms work very well to avoid a lost flow when a network load is at the designed level. The proposed approach can detect the links that make the network vulnerable to a more damaging failure under the current traffic demand pattern and adjust a flow so as to improve the survivability measure. When the load grows further, the attainable survivability measure depends on the amount of spare capacity installed in the network as well as the efficiency of the restoration schemes. The two-step restoration scheme is also shown to be very effective in promoting restorability after a failure. In the previous section, the three restoration schemes have been evaluated and compared in terms of required spare capacity cost. The results in this section give two additional factors in decision making on the selection of the fast

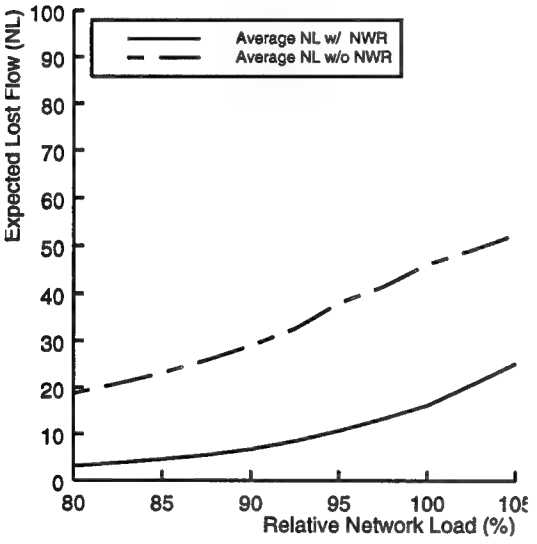
restoration schemes: restorability and applicability. The comparative analysis reveals that restorability largely depends on the regularity and connectivity of a network. Applicability of the MF-ETE scheme turns out to be somewhat restricted for a large network, and further speed-up of the solution procedure should be addressed in future research.



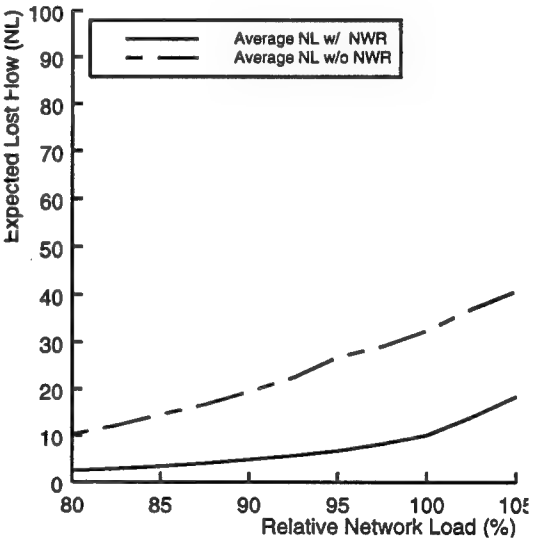
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network

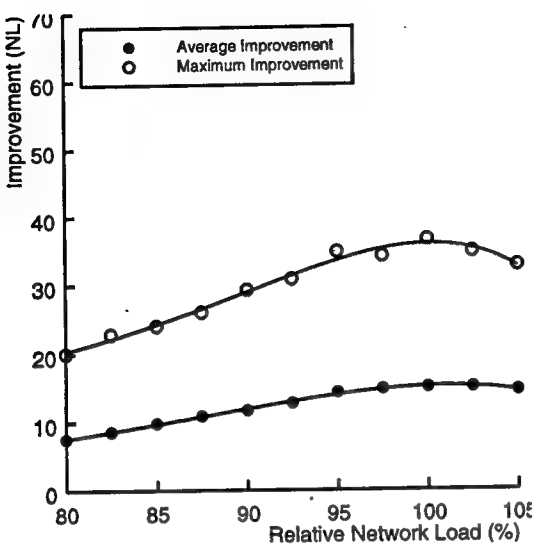


(c) (24,60) Sample Network

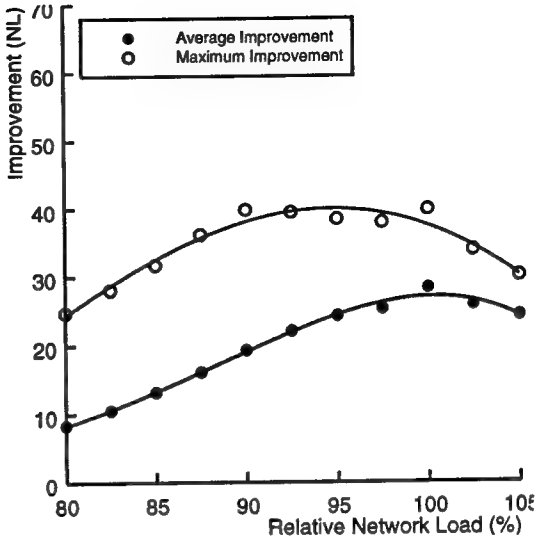


(d) US-WAN Sample Network

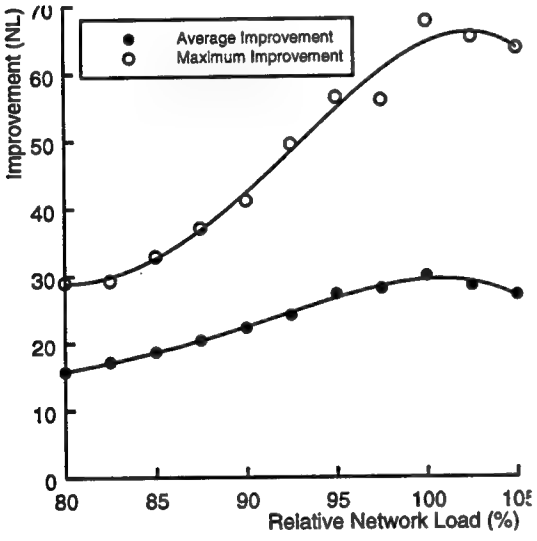
Figure 4-10. Attainable survivability measure with and without NWR (MF-LINE)



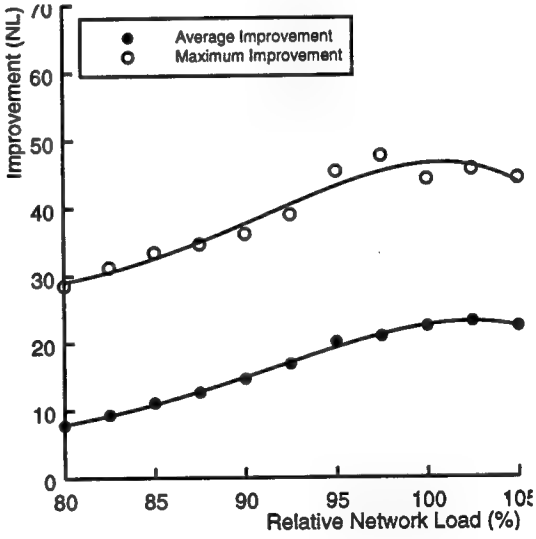
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network

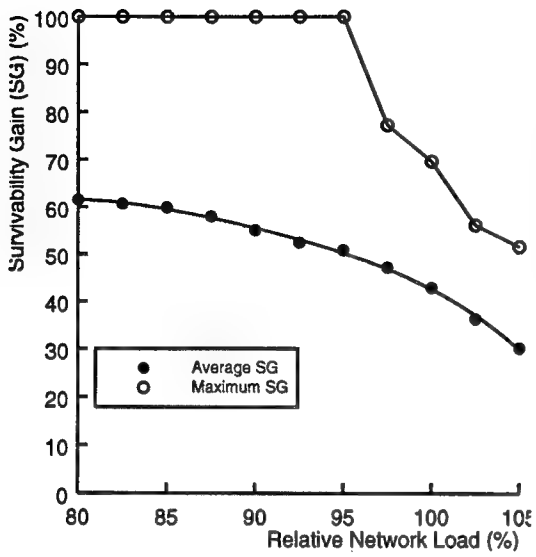


(c) (24,60) Sample Network

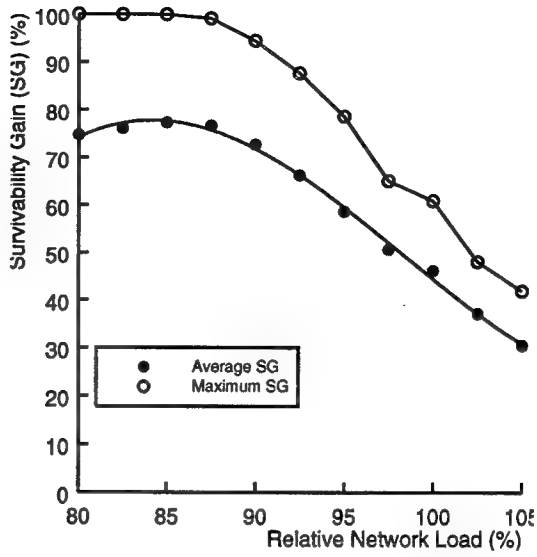


(d) US-WAN Sample Network

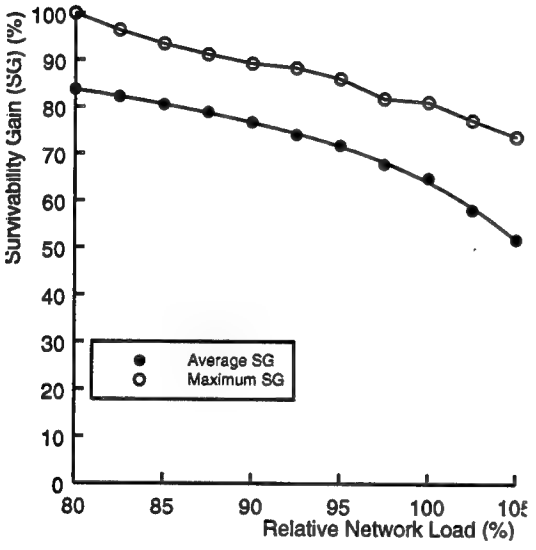
Figure 4-11. Improvement of survivability measure with NWR (MF-LINE)



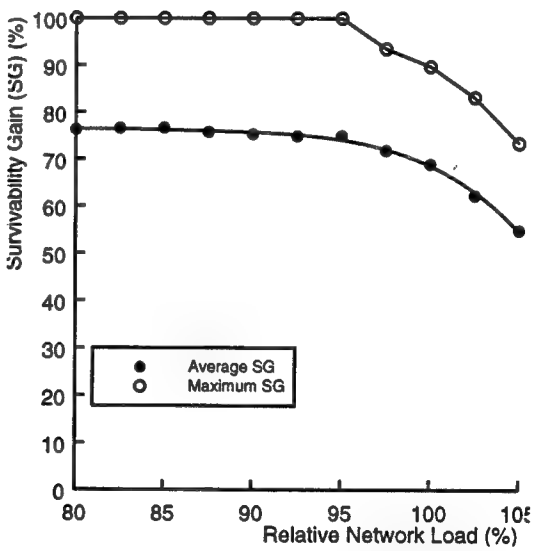
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network

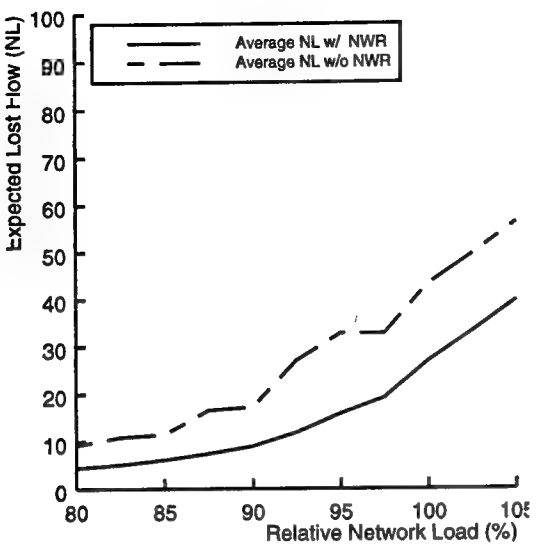


(c) (24,60) Sample Network

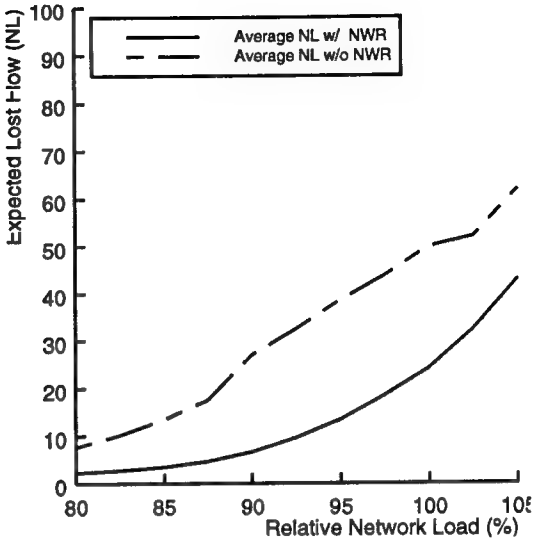


(d) US-WAN Sample Network

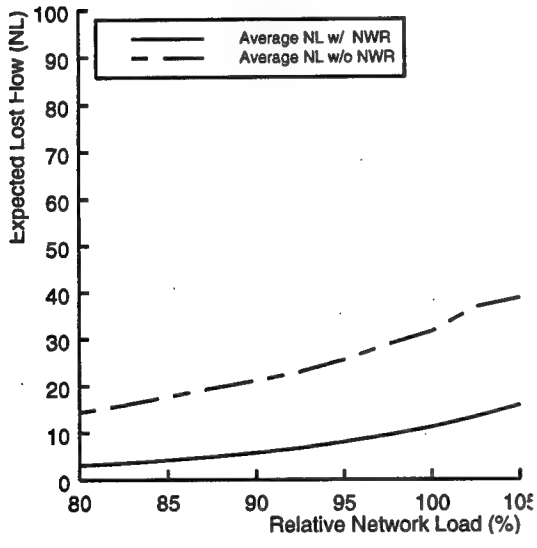
Figure 4-12. Survivability gain due to NWR (MF-LINE)



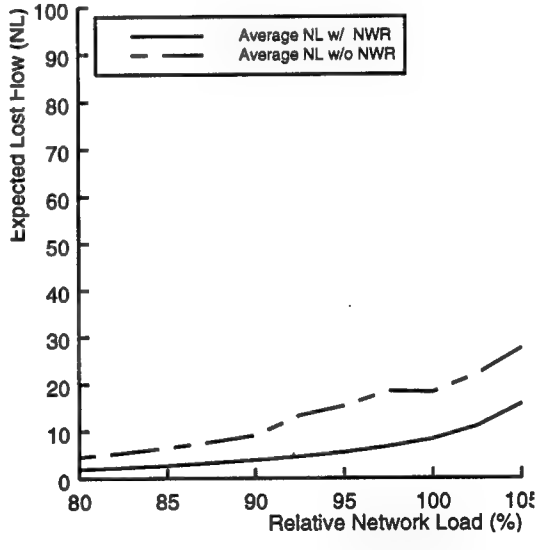
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network

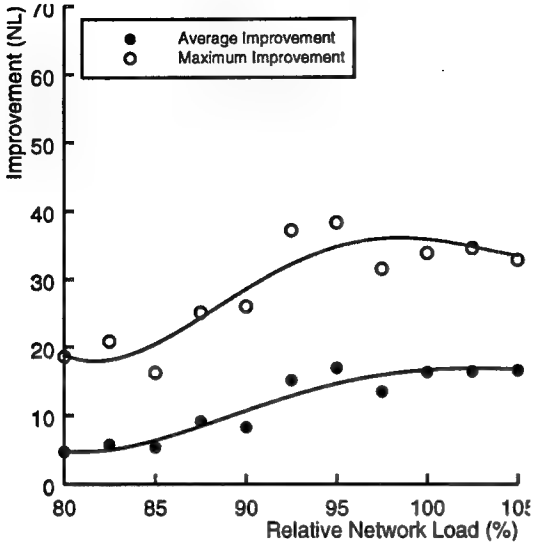


(c) (24,60) Sample Network

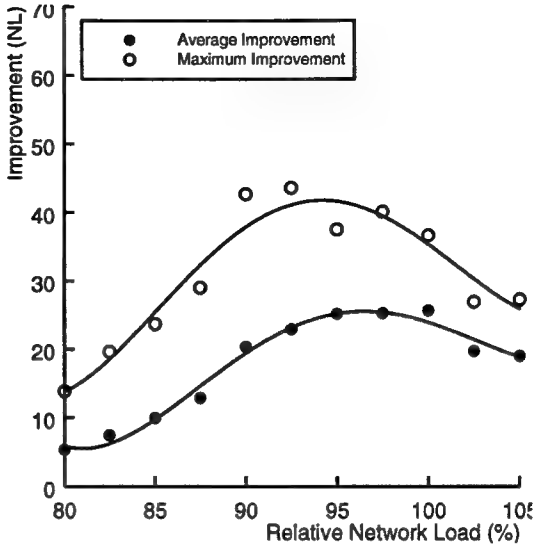


(d) US-WAN Sample Network

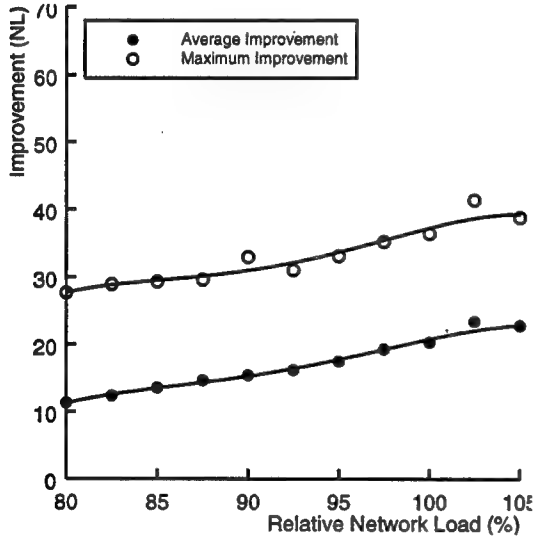
Figure 4-13. Attainable survivability measure with and without NWR (KSP-LINE)



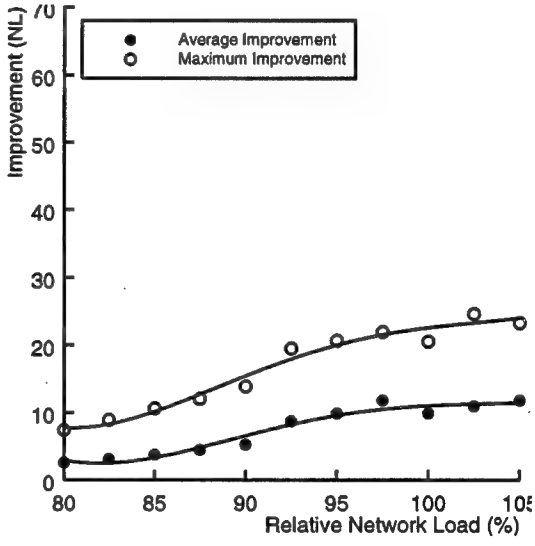
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network



(c) (24,60) Sample Network



(d) US-WAN Sample Network

Figure 4-14. Improvement of survivability measure with NWR (KSP-LINE)

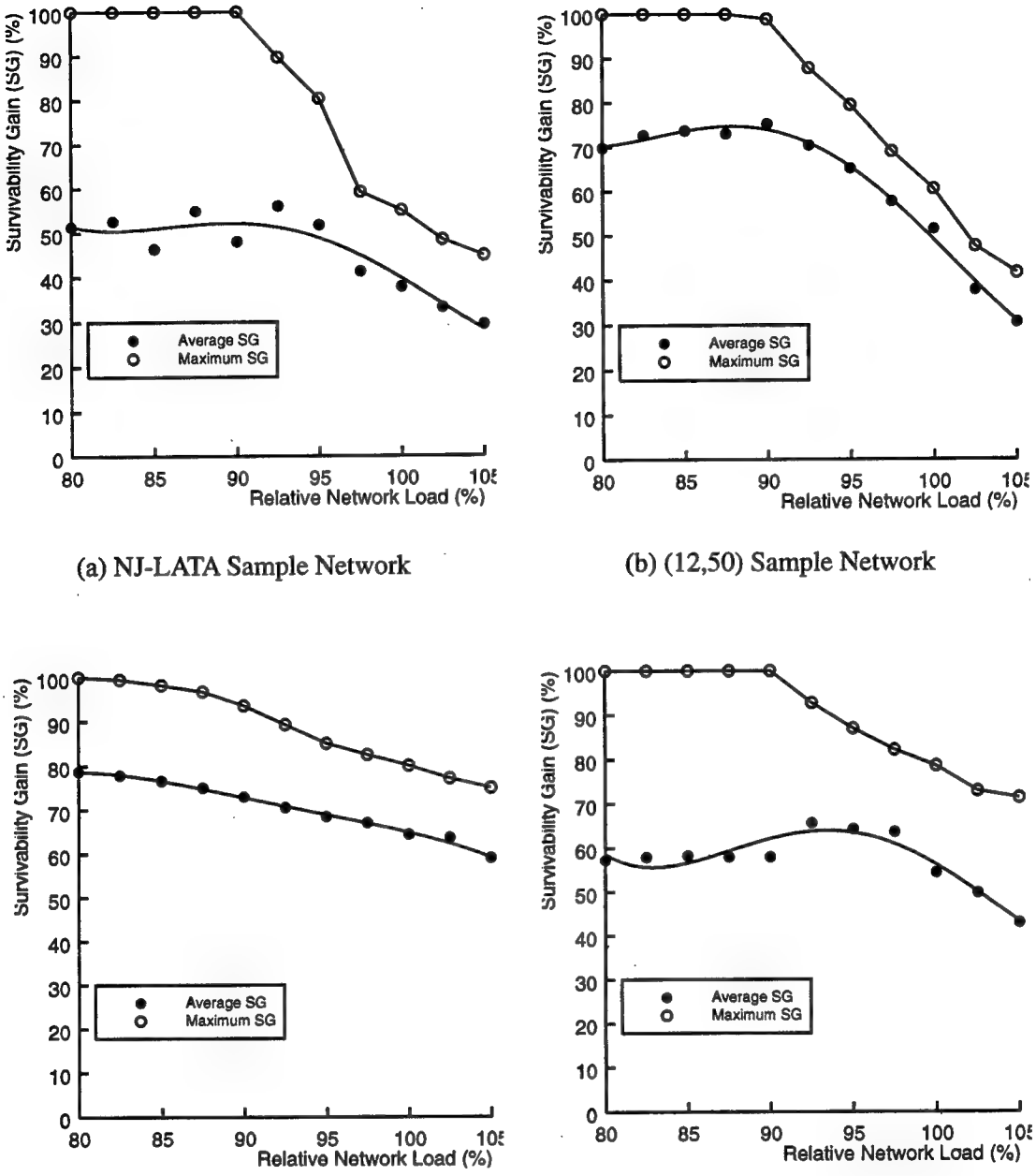
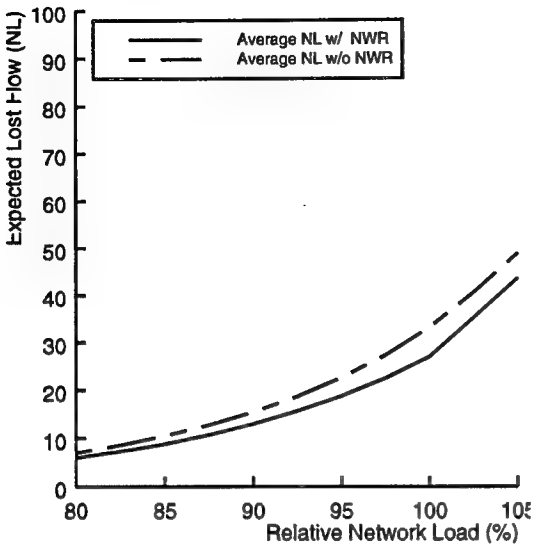
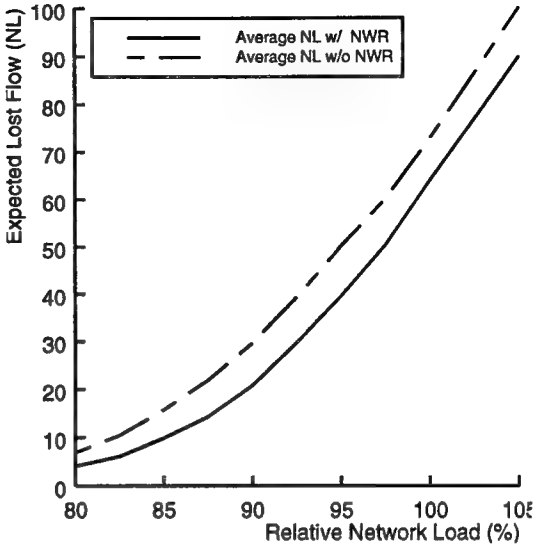


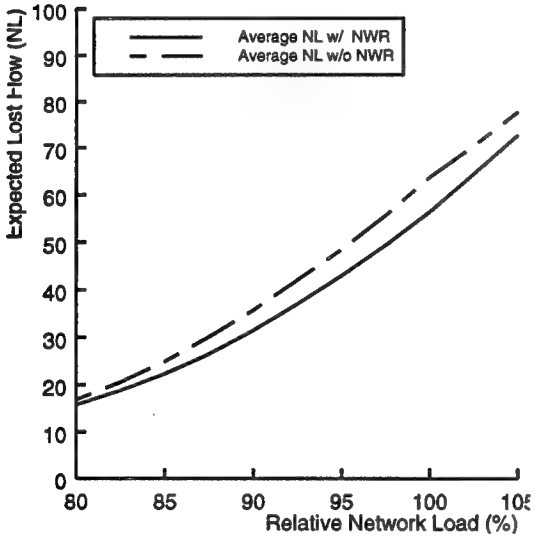
Figure 4-15. Survivability gain due to NWR (KSP-LINE)



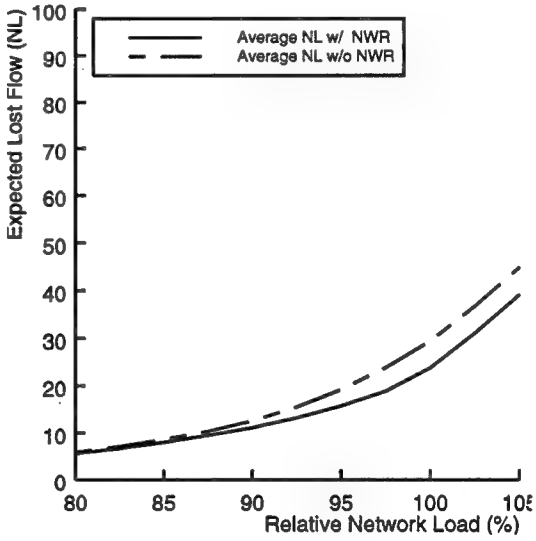
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network

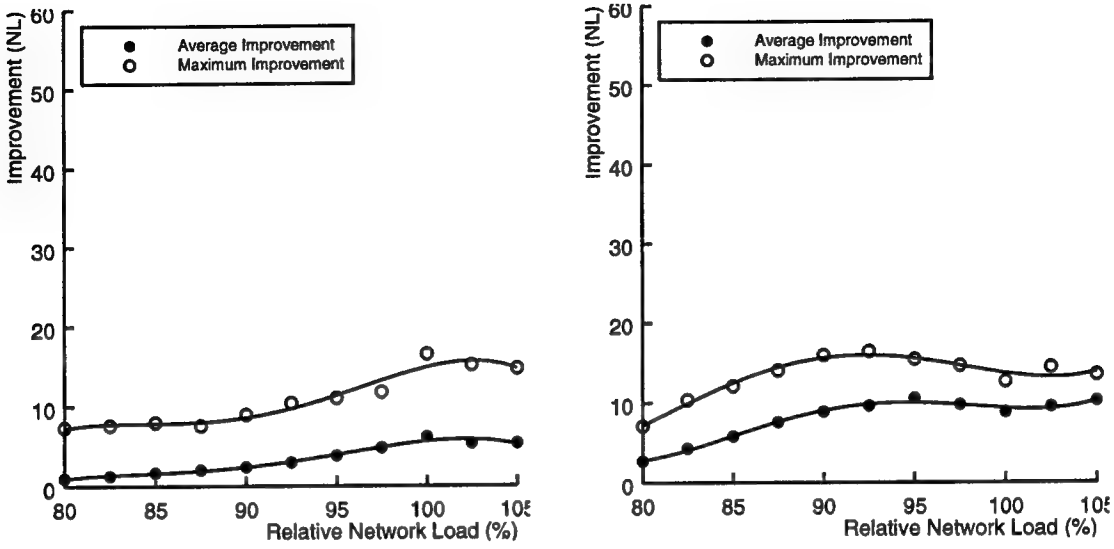


(c) (24,60) Sample Network



(d) US-WAN Sample Network

Figure 4-16. Attainable survivability measure with and without NWR (MF-ETE)



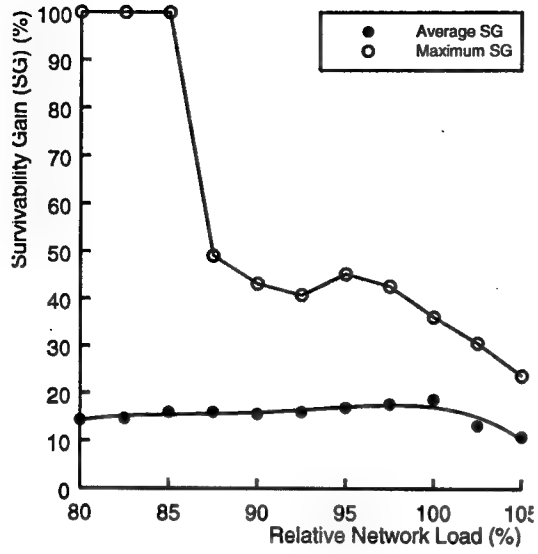
(a) NJ-LATA Sample Network

(b) (12,50) Sample Network

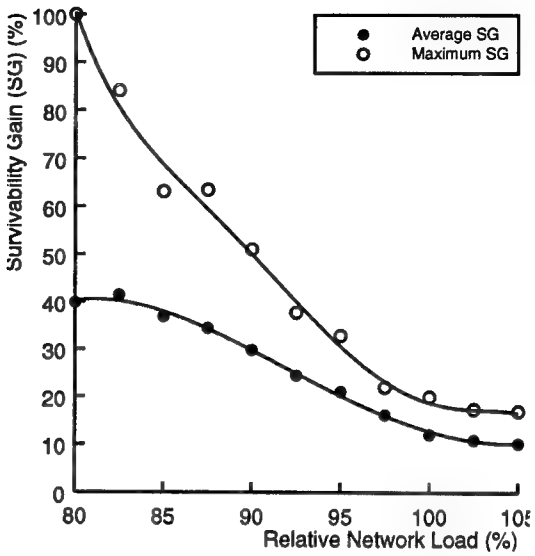
(c) (24,60) Sample Network

(d) US-WAN Sample Network

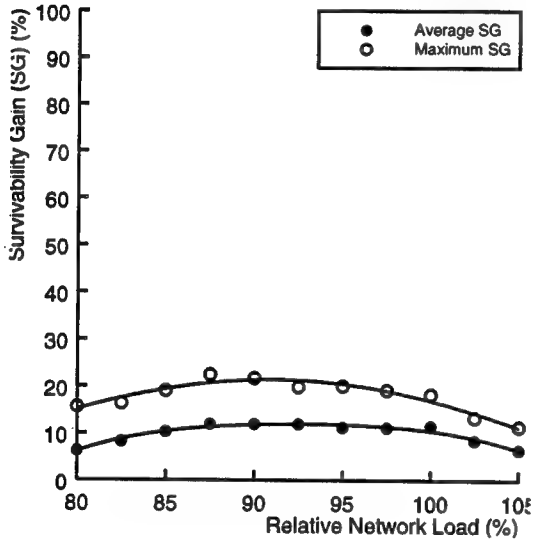
Figure 4-17. Improvement of survivability measure with NWR (MF-ETE)



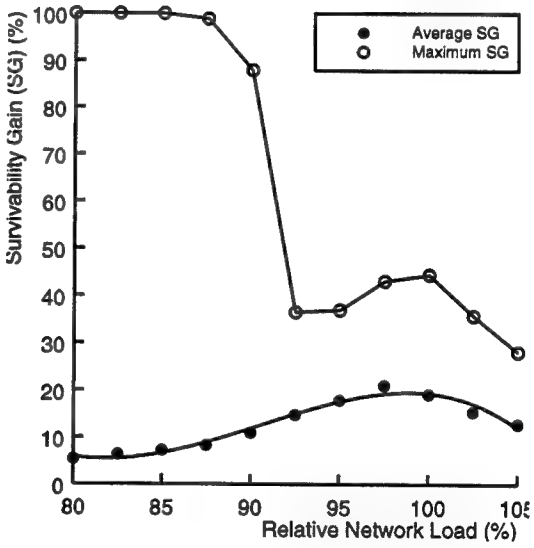
(a) NJ-LATA Sample Network



(b) (12,50) Sample Network



(c) (24,60) Sample Network



(d) US-WAN Sample Network

Figure 4-18. Survivability gain due to NWR (MF-ETE)

V Conclusion

This concluding section summarizes the major contributions of this work and suggests possible directions for future research.

5.1. Summary of Major Contributions

The significant contributions of this project are accomplished in the field of survivable network management for high-speed ATM inter-office networks. The project has provided an integrated view of survivable ATM network management systems with fast virtual path restoration capabilities. The two concepts, two-step restoration and survivable ATM network management architecture, have been introduced. We have identified two key open issues, the SCFA and SVPR problems, and have developed the optimization models and the solution procedures for the problems. The proposed algorithms have been applied to comprehensive comparative analysis on the fast restoration protocols. Figure 5-1 summarizes these contributions.

The first major contribution of this project is the identification of three key issues for survivable ATM network management and their consolidation in order to deploy an efficient and cost-effective survivable ATM networks. The identified issues are the fast restoration scheme, the survivable virtual path routing (SVPR) problem and the survivable capacity and flow assignment (SCFA) problem. The fast restoration mechanism performs run-time rerouting of affected virtual paths to provide service continuity, while the SVPR problem aims at maximizing restorability through dynamic virtual path reconfiguration in response to demand dynamics. The SCFA problem designs a fully restorable facility network with minimum cost, with the aid of a joint optimization procedure. The restorability of a fast restoration scheme completely depends on network design and virtual path flow assignment. Therefore, these three key issues should be addressed in an integrated manner to effectively realize survivable networks, but all previous works have overlooked addressing the issues in this way. These survivability functions are integrated into an existing ATM network resource architecture, leading to the survivable ATM network management architecture

proposed in this project. The complexity of the resource management process is simplified by classifying different levels of network resources and traffic entities into layers. Survivability functions are effectively consolidated into the virtual path layer and facility network layer.

Another fundamental contribution of this project is the introduction of the two-step restoration concept in order to enhance an attainable survivability level after a failure. It accommodates two

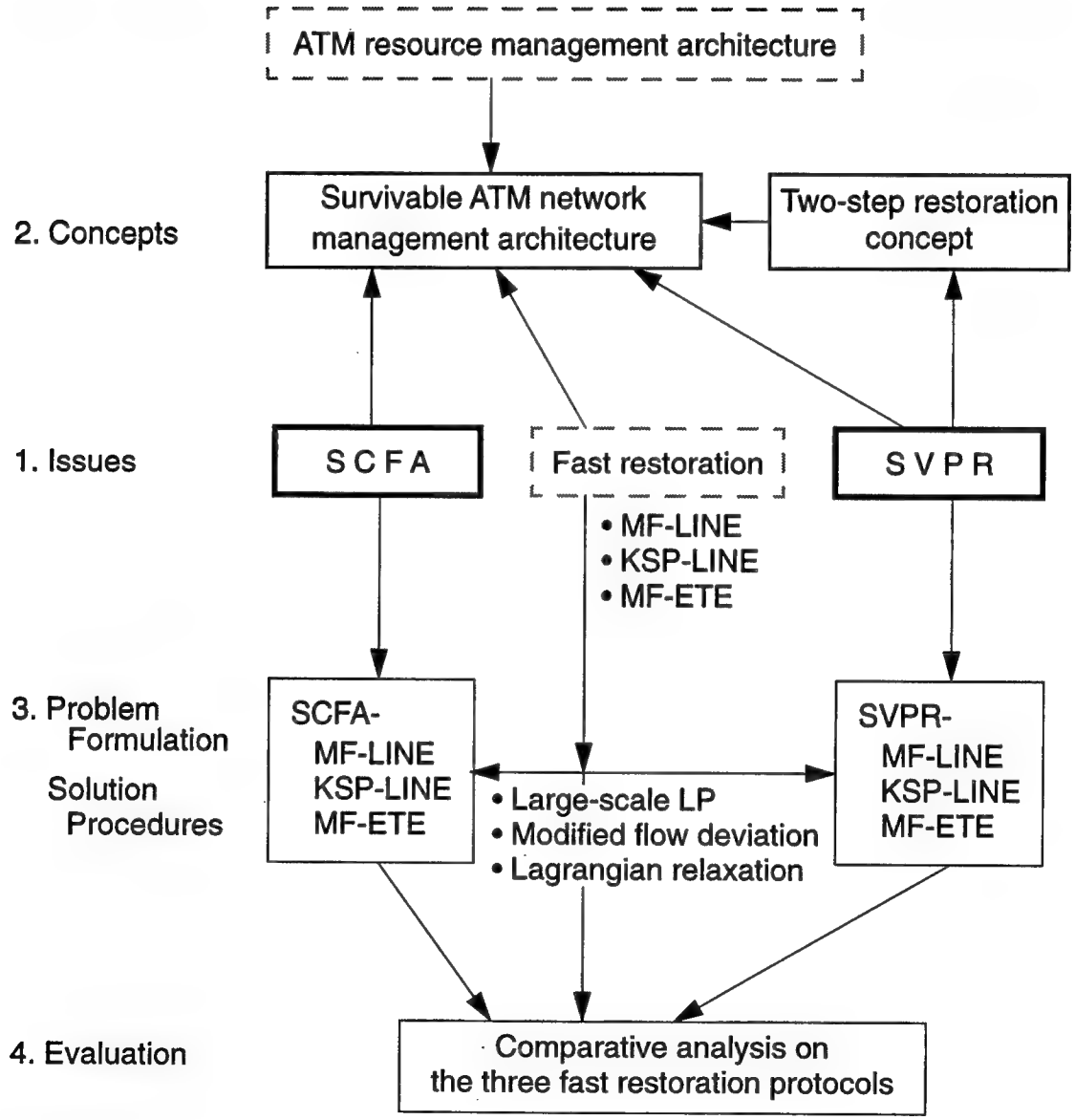


Figure 5-1. Major contributions of the project

contradicting requirements when a network failure happens: fast restoration and optimal virtual path reconfiguration. The scheme has been proven to effectively prevent lost flow due to a subsequent failure, especially in line restoration-based networks.

Optimization models have been developed for the SCFA and SVPR problems. The problem formulation differs depending on the fast restoration scheme employed in the network. Three representative fast restoration schemes have been identified, and distinct solution procedures have been proposed accordingly. The problems can be formulated as a large scale linear programming problem for MF-based networks. In order to overcome a huge number of constraints inherent to the problem, several mechanisms have been utilized, such as a column generation technique and arc-chain flow representation which help to reduce the number of constraints to a manageable size. Out of the employed techniques, the original ideas are to notice the basis matrix arrangement and to develop the direct $\hat{L}\hat{U}$ submatrix update mechanism. The identified basis matrix arrangement facilitates the LU decomposition of the basis matrix, and its validity has been proven in the project. These two mechanisms significantly economize the computation of the algorithm. Row generation and deletion mechanisms have been further devised to cope with the explosive number of constraints for the ETE-based networks.

The SVPR-KSP-LINE problem has been formulated as a non-differentiable multicommodity flow problem with linear constraints. The modified flow deviation method is another significant contribution of this project. The premature convergence around kinks which is inherent to non-smooth optimization could be avoided by properly adjusting optimization parameters. It has been shown that the procedure quickly converges to a near-optimal solution. Further lost flow prevention could be attained by a variation of the Lagrangian relaxation technique proposed in this project, but it comes at the expense of the computational time. This procedure also helps in finding a tighter lower bound estimate.

Finally, the performance evaluation of the proposed algorithms has not only verified the effectiveness of the proposed survivable management systems, but also has given an insight into the benefit of each restoration scheme. The economical benefit due to the end-to-end restoration scheme has been especially investigated, and the effect of network topology has been examined. Previous comparative studies have only been made qualitatively or through a simulation study. With the proposed optimization procedures, we have clarified for the first time their pros and cons

in terms of the optimal capacity installation cost. Contrary to a wide belief in the economic advantage on the end-to-end restoration scheme, our comprehensive study has revealed that attainable gain could be nominal for well-connected and/or unbalanced networks.

5.2. Future Extensions

This is the first work on survivable ATM network management where fast restoration mechanisms are taken into account. We have addressed the most fundamental and important aspects of the problem in this project. However, there are still several open issues that require further research. We conclude the project by attempting to identify some important issues to be addressed as an extension of the research.

- *Topological effect and topological design*: Further investigation would be necessary to clarify the effect of network topology on the required spare capacity cost as well as restorability. This study would be useful for a topological design which is required to construct a new transmission network.[1]

- *Cost function*: In the SCFA problem, we assume a linear cost function. Its major benefit is the substantial simplification of the solution procedure, which accounts for the popularity of the linear approximation in most previous works. However, other cost functions might be preferable if more precise approximation of the installation cost is necessary. For example, a logarithmic cost function can accommodate the following effect of large-scale economy: As the channel size increases, the incremental cost per bit decreases [44].

- *Failure type*: In this work, a complete span cut has been considered as the most plausible failure event, and the survivable network management scheme has been developed based on this assumption. Other types of failure scenarios should be taken into account to enhance the network survivability, if their likelihood is not small enough to neglect. Out of such candidates, a node failure would be the most important event to consider in future research.

- *Speed-up*: Further work is required in the development of the SVPR-MF-ETE solution procedures in order to complete within a reasonable computation time at a higher load. Heuristics or polynomial time linear programming techniques [42] might be investigated.

1. When applied to an existing network where a network topology is given, however, the design approach developed in this project would suffice. In the design of a survivable network management system, a fast restoration scheme should be chosen with respect to the economy, restorability, applicability, restoration speed and control complexity. The proposed solution procedures would help this decision-making process in terms of the first three criteria.

- *Lower level QOS*: When a load is less than the designed level, a solution to the SVPR problem is not usually a singleton. Although any solution could be selected from a survivability viewpoint, a different VP assignment could lead to a different quality of service at a lower layer. Our approach, of course, guarantees that the QOS exceeds the designated level by satisfying the VP-level traffic demand. However, it would be desirable to have some mechanism which selectively chooses a solution from the optimal set, since this could further improve the QOS at call and cell layers.

- *Initial feasible solution to the SVPR problem*: It is possible that no feasible solution would exist for a given traffic demand. In such a case, a part of the demand must be somehow rejected. For example, we must find a flow assignment which minimizes the amount of rejected flow[2]. Note that although this precaution must be installed in the network, the above situation would seldom happen if a network is properly designed. Significantly more redundant capacity is required in order to assure full restorability against any link failure for a projected traffic demand. Therefore, a feasible flow can be obtained unless considerably heavier traffic demand is offered to the network than the projected level or unless multiple network failures are present. We have not encountered such a situation in our experiments where the load is increased up to 25 percent more than the projected level. Since demand projection is based on the busy hour traffic for the forthcoming design period (say a year), a load should be typically 100% RNL or less in normal operating conditions. Furthermore, in long-haul or regional exchange networks, where precautions against a network failure are highly necessary, a network load is typically smoothed out since a large amount of traffic is integrated there. Therefore, 125% or more RNL would arise only by overshooting demand across the entire network, which would seldom happen in a practical situation.

- *Transient phenomena after a failure*: Based on the fast restoration scheme, the effect of a failure would be confined to affected virtual paths. However, if excessive data retransmission is necessary, it might also affect other traffic along the restoration routes over some duration after a failure. This would happen if data traffic dominates over the network. One possible solution is to increase the required bandwidth over restoration paths in order to accommodate effective bandwidth during this congested period. Further investigation would be necessary to determine the (VP-level) equivalent bandwidth on such an occasion.

---

2. We can formulate this problem as an LP problem.

- *Multiple projected demand patterns*: This project considers a single projected traffic demand pattern in the network synthesis problem. However, it would be preferable to consider multiple projected traffic demands in a certain situation. For example, such a provision would be appropriate for a network covering a wide area with multiple time zones because it creates multiple busy hour traffic patterns. Since this would significantly increase the dimension of the problem, further precautions should be devised to solve the problem in a reasonable time.
- *SCFA-KSP-LINE*: A heuristic approach has been applied to the SCFA-KSP-LINE problem. Further research is required to find a better solution in this problem. A Lagrangian relaxation technique might be also applied to this problem.

Appendix A. *SCFA-MF-LINE:* Problem formulation

The following notations are used in the formulation of the SCFA-MF-LINE problem[1].

P^{π} : A set of all possible routes for commodity $\pi \in \Pi$.

RP^{a} : A set of all possible restoration routes for arc $a \in A$.

$\underline{x} = (x_p^{\pi})$ $(\forall \pi \in \Pi, \forall p \in P^{\pi})$: A commodity path flow vector.

$\underline{r} = (r_p^{a})$ $(\forall a \in A, \forall p \in RP^{a})$: An arc restoration path flow vector.

$\underline{z} = (z_a)$ $(\forall a \in A)$: A spare capacity vector.

d_a $(\forall a \in A)$: A unit arc installation cost. It is assumed to be non-negative.

$E_a \equiv \{l \in E : a \notin l\}$: A set of all links except for the one containing arc a.

$\theta_{a,p}$: An arc-path indicator variable which equals 1 if arc a is contained in path p, and 0 otherwise.

f_a $(\forall a \in A)$: The amount of a flow of arc a.

z_a^l $(\forall a \in A, \forall l \in E_a)$: The amount of necessary spare bandwidth of arc a upon a failure of link l.

σ^{π} $(\forall \pi \in \Pi)$: The price of commodity (dual variable).

w_a $(\forall a \in A)$: The price of arc restoration (dual variable).

μ_a^l $(\forall a \in A, \forall l \in E_a)$: The price of arc per failure (dual variable).

Assuming that the network resource installation cost is a linear function of arc capacity, the

1. Refer to Appendix T for the complete listing of the notations used in the report.

SCFA-MF-LINE problem becomes a linear programming (LP) problem. The arc-chain flow representation [9] is employed on the commodity flow as well as the restoration flow[2]. The SCFA-MF-LINE problem and its dual problem (SCFA-MF-LINE-DUAL) are formulated as follows:

Minimize $D(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{z}) = \sum_{a \in A} d_a \cdot c_a$ **(SCFA-MF-LINE)**

over $\underset{\sim}{x} \geq 0, \underset{\sim}{r} \geq 0, \underset{\sim}{z} \geq 0$

subject to a) $\sum_{p \in P^\pi} x_p^\pi = q^\pi \qquad \forall \pi \in \Pi$ *(A-1)*

b) $z_a - z_a^l \geq 0 \qquad \forall a \in A, \forall l \in E_a$ *(A-2)*

c) $\sum_{p \in RP^a} r_p^a - f_a = 0 \qquad \forall a \in A$ *(A-3)*

$c_a = f_a + z_a$

$f_a = \sum_{\pi \in \Pi} \sum_{p \in P^\pi} \theta_{a,p} \cdot x_p^\pi$

$z_a^l = \sum_{b \in l} \sum_{p \in RP^b} \theta_{a,p} \cdot r_p^b$

Maximize $\sum_{\pi \in \Pi} q^\pi \cdot \sigma^\pi$ **(SCFA-MF-LINE-DUAL)**

over σ^π, w_a unrestricted, $\mu_a^l \geq 0$

subject to i) $\sum_{a \in p} (d_a + w_a) \geq \sigma^\pi \qquad \forall p \in P^\pi, \forall \pi \in \Pi$ *(A-4)*

ii) $\sum_{b \in p:\,(l \ni a)} \mu_b^l \geq w_a \qquad \forall p \in RP^a, \forall a \in A$ *(A-5)*

---

2. There are two ways to represent a flow in a network. The *arc-node flow representation* uses a collection of arc flows to represent a network flow, while the *arc-chain flow representation* employs a set of path (and cycle) flows [4] [9].

iii) $$\sum_{l \in E_a} \mu_a^l \le d_a \qquad \forall a \in A \qquad (A\text{-}6)$$

$$\mu_a^{l:(a \in l)} \equiv 0$$

The flow conservation law (A-1) says that the traffic demand of a commodity is satisfied over virtual paths for the commodity. The capacity constraints (A-2) state that the spare capacity of each arc must be at least equal to the amount of the arc bandwidth required for restoration from any single-link failure. The constraints are expressed for every combination of a link failure event and a remaining arc after the failure. The full restorability constraints (A-3) enforce restoration of all affected arc flow over the restoration paths. σ^π, μ_a^l and w_a are the simplex multipliers corresponding to the constraints (A-1), (A-2) and (A-3), respectively.

Appendix B. SCFA-MF-LINE: Solution approach

The SCFA-MF-LINE problem is modeled as a large-scale LP problem with a special structure. Since direct application of an LP algorithm to this problem requires highly intensive computation, the following modifications are made. First of all, the *arc-chain flow representation* is selected instead of the arc-node representation [9]. The arc-chain flow representation significantly lowers the size of a constraint set, especially for a large network. For example, the size of flow conservation constraints is reduced from $|\Pi| \cdot |V|$ $(O(V^3))$ to $|\Pi|$ $(O(V^2))$. For example, 1,210 constraints is reduced to 110 constraints for an NJ-LATA network[3] and from 21,168 constraints to 756 constraints for a US-WAN network[4].

The *column generation technique* is applied to accommodate infinitely many path variables in the arc-chain flow representation [50]. The technique generates variables as needed during the course of the algorithm instead of listing all columns at once. This strategy utilizes the fact that only a few paths will be actually used in the optimal solution. The solution procedure is decom-

3. A network with n nodes and m arcs is referred to as an (n, m) network in this report.
4. It is assumed that one commodity is defined for each source and destination node pair in this calculation.

posed into a master process and a sub-process. The sub-process adopts the simplex algorithm and calculates an optimal solution using generated columns. On the other hand, the master process checks whether the obtained solution is globally optimal or not. The global optimality is verified if the dual feasibility conditions (A-4) and (A-5) hold[5]. Although the conditions must be satisfied for all commodity paths and all restoration paths, it suffices to examine their shortest paths. The condition (A-4) is satisfied if the length of the shortest commodity path of π under modified arc length $(d_a + w_a)$ is not less than σ^{π}. The condition (A-5) holds if the length of the shortest restoration path of arc a under metric $\{\mu_b^l : a \in l, b \in A\}$ is not less than w_a. Dijkstra's algorithm can be applied to find the shortest paths for each commodity and for each arc. If the dual feasibility conditions are violated for any shortest path, then the total capacity installation cost could be reduced by using this path. Therefore, the master process generates all violated shortest path variables[6] and invokes the sub-process. Otherwise, the obtained solution is globally optimal. After the completion of the procedure, a post-processing routine is called to make the link bandwidth a multiple of that of available optical fiber cables (round-up procedure). The proposed solution procedure is summarized in Figure B-1. The initialization process generates only the columns pertaining to the shortest commodity paths and the shortest restoration paths and obtains an initial feasible solution.

The revised simplex method has been widely used due to its effectiveness in reducing both running time and memory space [17] [54], and it is applicable to the sub-process. Each iteration of the revised simplex method requires solving two systems of linear equations to find an entering column vector and a set of simplex multipliers [17]. The basis matrix is factorized into an *LU* form in order to facilitate the solution of the linear systems[7]. The computational load of the revised simplex method is dominated by the update of the *LU* matrices and the solution of two linear equations. Thus, the dimension of the basis matrix largely determines the speed of the algorithm. In the SCFA-MF-LINE problem, the dimension can be to the order of 10^3 or more for a large network, even with the reduction introduced by the arc-chain flow representation. Since the required computation for the revised simplex method becomes nontrivial in such a large-scale LP problem, it is necessary to exploit the structure of the problem in order to reduce the computational burden. The following observations are used to develop our algorithm:

---

5. Note that the condition (A-6) and non-negativity of μ_a^l are assured at the end of the sub-process.

6. At the end of the sub-process, the dual feasibility conditions are satisfied for all generated path variables. Thus, if the condition is violated for a path at the master process, then this path has not yet been generated.

7. This approach has also been shown to possess numerical stability [54].

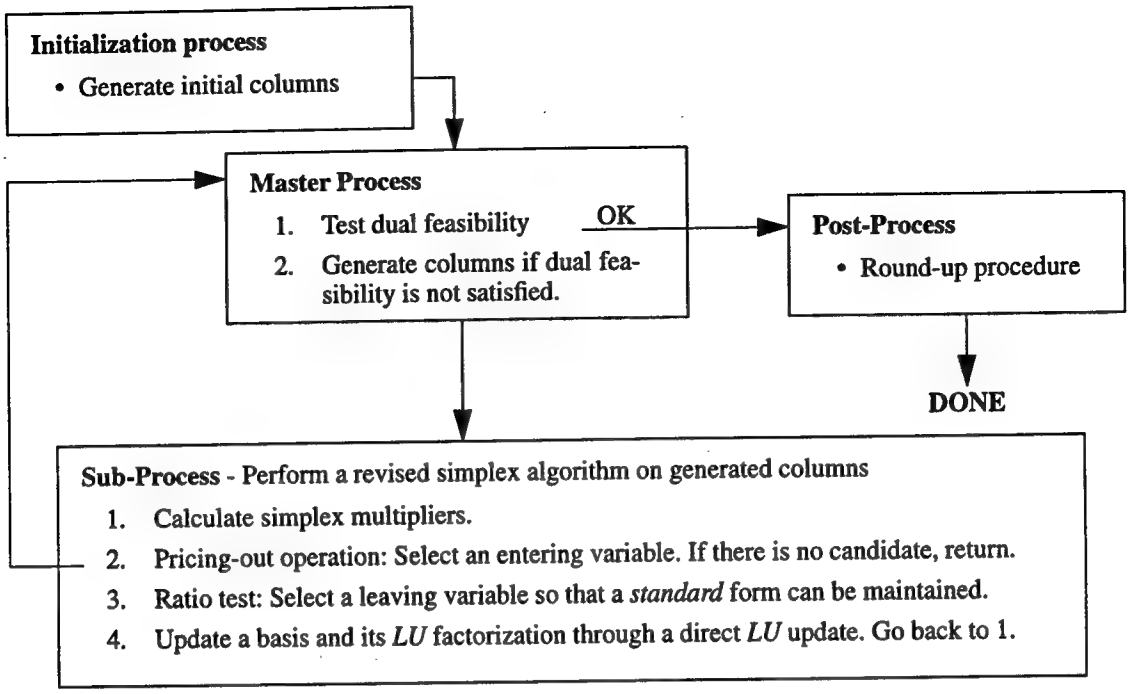


Figure B-1. Solution procedure for the SCFA-MF-LINE problem

At the end of any iteration, the following two statements are true if $q^{\pi} > 0$ for $\forall \pi \in \Pi$:

1. For each commodity $\pi \in \Pi$, at least one commodity path flow variable, x_p^{π} ($p \in P^{\pi}$), is in the basis.
2. For each arc $a \in A$, it is possible to maintain at least one restoration path flow variable, r_p^a ($p \in RP^a$), in the basis.

We randomly choose one such basic variable from each commodity and each arc, and call it a *key* flow variable. At each iteration the basis matrix B can be arranged as follows:

$$B = \begin{bmatrix} I & & & C \\ R & I & & D \\ & H_1 & -I & K_1 & M_1 \\ & H_2 & & I & M_2 \\ & H_3 & & K_3 & M_3 \end{bmatrix} \qquad (B\text{-}1)$$

where I is an identity sub-matrix. The first set of columns corresponds to the key commodity path flow variables, and the second set is composed of the key restoration path flow variables. The

remaining commodity and restoration path variables are collected into the last set. The third and fourth sets of columns contain the slack variables of the capacity constraints and the spare capacity variables, respectively. The first set of rows represents the flow conservation law (Equation (A-1)), and the second set represents the full restorability constraints (Equation (A-3)). The capacity constraints (Equation (A-2)) are collected and arranged in the last three sets of rows so that the identity submatrices with proper sign can appear in the place shown in Equation (B-1). All submatrices, R, H_i, K_i, M_i (i=1,2,3) are sparse. The basis matrix arranged in the above manner is called a *standard* form in this project.

Using the above matrix arrangement, the LU matrices can be readily obtained as follows:

$$B = \begin{bmatrix} I & & & \\ R & I & & \\ & H_1 & I & \\ & H_2 & & I \\ & H_3 & K_3 & \hat{L} \end{bmatrix} \cdot \begin{bmatrix} I & & & C \\ & I & & F \\ & & -I & K_1 & Y_1 \\ & & & I & Y_2 \\ & & & & \hat{U} \end{bmatrix}$$

where $F = D - RC$, $Y_i = M_i - H_i \cdot F$ for $i = 1, 2, 3$ and $Z \equiv \hat{L} \cdot \hat{U} = Y_3 - K_3 \cdot Y_2$. The major computational task here turns out to be the factorization of the submatrix Z since the matrix multiplications to obtain F, Y_i and Z can be easily performed by exploiting the problem structure. The dimension of Z is identical to the number of non-key flow variables which is significantly smaller than that of the original matrix, B. According to our computational experiments, the dimension of Z is only up to an eighth to a fourth of dim(B), resulting in a large reduction of memory space and execution time. The algorithm runs about 30 times faster than the method which factorizes the basis matrix at each iteration[8].

By taking advantage of the similarity between two successive submatrices, we can further economize the computation. The $\hat{L}\hat{U}$ submatrices can be directly updated instead of factorized from scratch at each iteration. Depending on the type of the entering and leaving variables, the pivoting operation can be categorized into four classes:

Class 1)The entering and leaving variables are both slack variables, and the capacity constraint corresponding to the entering slack variable is in the fifth set of rows. In this case, a

8. This speed-up is based on the experiments on the NJ-LATA sample network shown in Figure 3-4, "Sample networks 1. Real networks," on page 41. Further speed-up is expected for a large network.

row of H_3, K_3 and M_3 changes in the same position, resulting in a single row update in the submatrix Z (Figure B-1-a).

Class 2)A commodity or restoration path flow variable enters and a non-key flow variable leaves. In this case, the new submatrix Z differs from the previous one only in a single column because a column update of C, D, M_1, M_2 and M_3 occurs in the same position (Figure B-1-b).

Class 3)The entering variable is a flow variable and the leaving variable is a slack variable. The dimension of the submatrix increases by one in this case. The update can be arranged so that a new column is added to C, D, M_1 and M_2 in the last column. A new row is appended to H_3 and K_3 in the last row, and a new row and a new column are added to the submatrix M_3 in the corresponding position. This results in attaching a single row and column to the previous submatrix Z (Figure B-1-c).

Class 4)In other cases, no explicit rule exists to realize a direct $\hat{L}\hat{U}$ update.

A direct $\hat{L}\hat{U}$ update is possible for the first three classes, and further improvement on the computation time will be attained if these three types of pivoting are dominant. Table B-1 shows the frequency of pivoting for each class based on our experiments. More than 75 % of pivot operations fall into the above three categories, and we could thus double the speed of the computation. The LU factorization routine is invoked at 18 to 25 percent of the iterations, and they are evenly distributed. Thus, the refactorization prevents a round-off error from propagating over the iterations and leads to a numerically stable operation.

The Bartels and Golub's decomposition method [12] [67] is applied in the first two classes. The $\hat{L}\hat{U}$ update procedures for Class 1 are summarized in the following passage. Class 2 can be handled in a similar fashion.

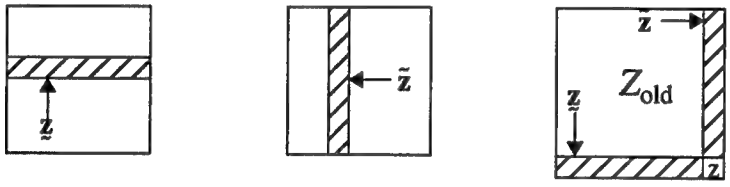
1. First, the new submatrix Z_{new} is factorized into $\tilde{L} \cdot \tilde{U}$ as shown in Figure B-2 (a), where L_{old} and U_{old} are the submatrices $\hat{L}$ and $\hat{U}$ before the basis update. $\tilde{U} = U_{old}$, while $\tilde{L}$ differs from L_{old} only by a single row as shown in the figure.
2. A cyclic row permutation on the matrix $\tilde{L}$ gives a lower Hessenberg matrix H as shown in Figure B-2 (b). In a matrix form, $H = P \cdot \tilde{L}$ where P is a permutation matrix.
3. Using elementary column operations, the matrix H can be transformed into a lower triangular matrix L_{new} as $L_{new} = \tilde{L} \cdot M$ where M is an elementary matrix.
4. An upper triangular matrix $U_{new} = M^{-1} \cdot \tilde{U}$.

Now the LU decomposition of the permuted submatrix $P \cdot Z_{new}$ is obtained as L_{new} and U_{new}. It may be necessary to interchange adjacent columns for pivoting during the third step, and this can be accomplished by a slight modification of the above procedure.

As for Class 3, the following procedures yield a new $\hat{L}\hat{U}$ decomposition:

1. First, the new submatrix Z_{new} is factorized into $\tilde{L} \cdot \tilde{U}$ as shown in Figure B-3 (c).
2. Using elementary row operations, $\tilde{U}$ can be transformed into an upper triangular matrix U_{new}. In a matrix form, $U_{new} = M \cdot \tilde{U}$ where M is an elementary matrix with a special structure as shown in Figure B-3 (d). No row permutation is necessary in this case.
3. A lower triangular matrix $L_{new} = \tilde{L} \cdot M^{-1}$. It has a closed form as shown in Figure B-3 (d)

Note that the worst case complexity of the proposed algorithm is exponential since the simplex algorithm is employed in the sub-process [17]. However, the running time is considerably reduced by the proposed mechanisms to a level where the algorithm is applicable to any network with a practically reasonable size. Based on our experiment using the DEC Alpha-station 200 4/233[9], the



a) Class 1 b) Class 2 c) Class 3

Figure B-2. The structure of a new submatrix Z

Class	min ~ max	average
Class 1	7.3 ~ 36.3%	(29.2%)
Class 2	17.5 ~ 46.0%	(26.8%)
Class 3	17.3 ~ 29.2%	(20.7%)
Class 4	17.5 ~ 25.3%	(23.3%)

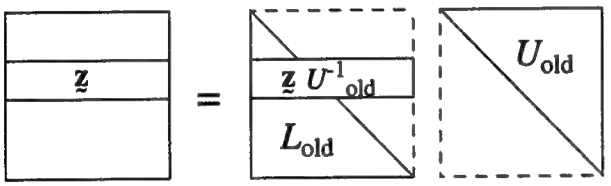
Table B-1. The frequency of pivoting by category

Two sample networks, the NJ-LATA and US-WAN networks, are used in the experiments (see Figure 3-4, "Sample networks 1. Real networks," on page 41). Ten randomly generated demands are employed for each network model.

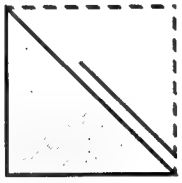
total computing time is less than 10 minutes for the NJ-LATA network and about an hour for the US-WAN network[10].

Appendix C. SCFA-MF-LINE: Validity of the algorithm

This section verifies the two claims which have been used in the development of our algorithm. The first claim is that a basis matrix can be arranged in a *standard* form in all iterations of the sub-process. The second claim is that a well-known shortest path algorithm can be applied to the dual feasibility tests in the master process. Note that the column generation algorithm guarantees the termination of algorithm at the optimum in a finite number of iterations if the generated columns

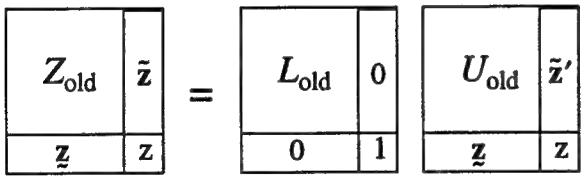


(a) $Z_{new} = \tilde{L} \cdot \tilde{U}$ in Class 1



(b) A lower Hessenberg matrix

$$\tilde{\mathbf{z}}' = L_{old}^{-1} \cdot \tilde{\mathbf{z}}$$



(c) $Z_{new} = \tilde{L} \cdot \tilde{U}$ in Class 3

(d) M and L_{new} for Class 3

Figure B-3. Direct *LU* update operations

9. This alpha station runs at 233 MHz (21064A) It has SPECint 157.6 and SPECfp 183.8. A 512 KB cache and a 64 MB memory are installed.

10. See Section 3.4.2. for details on the settings of the experiments.

are not removed [50].

Lemma C-1

Assume $q^{\pi} > 0$ for $\forall \pi \in \Pi$, and suppose that a positive flow is assigned for any arc in the algorithm. Then, a basis matrix B can be arranged as in Equation (B-1) in all iterative steps of the sub-process.

(proof)

Since $q^{\pi} > 0$ for $\forall \pi \in \Pi$, at least one commodity path flow variable must always be in the basis for each commodity. Similarly, based on the positivity assumption of an arc flow, at least one restoration path flow must be in the basis for each arc. Therefore, it is possible to choose a key flow column for each commodity and for each arc which constitutes the first and second sets of columns. The last set of columns has no restriction, and any non-key flow variables can be collected there in any order. Let s_a^l denote the slack variable of the capacity constraint for arc a against a failure of link l. The placement of all basic slack variables in the third set of columns requires that the corresponding capacity constraints be in the third set of rows. Now, place a basic spare capacity variable, z_a, in the fourth set of columns. This requires that some capacity constraint for arc a be arranged in the corresponding row in the fourth set. Therefore, a slack variable, s_a^l, must be non-basic for some $l \in E_a$. Assume otherwise. Suppose $\exists a \in A$ such that z_a and s_a^l for all $l \in E_a$ are in the basis. Let B_a be a sub-matrix of B composed of the columns z_a and s_a^l for $\forall l \in E_a$. Although B_a is composed of $|E_a| + 1$ columns, $\dim(B_a) = |E_a|$. Thus, the column vectors of B_a are linearly dependent. This contradicts the fact that the column vectors of B_a are a part of the basis.

(Q.E.D.)

Note that the first assumption is not restrictive since a commodity with no demand can be eliminated from the set Π in the problem formulation. The second assumption requires that any arc has a positive flow, which usually holds for any properly designed network. The assumption is violated only if an arc is unnecessarily installed between nodes where no load is offered due to a lack of demand. Furthermore, the following Lemma suggests that it is possible to maintain at least one restoration flow variable in the basis even if an arc flow becomes zero at some point in the sub-process. Consequently, the second assumption is not necessary.

Lemma C-2

Suppose that r_p^a is the only restoration flow variable for arc a in the basis, and it is selected as a leaving variable at an iteration in the sub-process. Furthermore, suppose that the entering variable is not a restoration flow variable for arc a. Then there always exists another candidate than r_p^a that will leave the basis.

(proof)

Let B be the current basis matrix before pivoting, m be the dimension of B ($m = |B|$), and n be the number of all generated variables. Suppose that the q-th column vector ($q > m$) is selected as an entering variable, and the leaving variable, r_p^a, is the r-th column vector ($r \le m$) before pivoting. The following notations are used throughout the proof.

$\underset{\sim}{a}_q = (a_{iq})$ $(i = 1, \ldots, m)$: The column vector corresponding to the entering variable.

$\underset{\sim}{b} = (b_i)$ $(i = 1, \ldots, m)$: The right-hand-side column vector.

$\underset{\sim}{a}_q' = (a_{iq}')$ $(i = 1, \ldots, m)$: $\underset{\sim}{a}_q' \equiv B^{-1} \cdot \underset{\sim}{a}_q$ where B is the basis matrix before pivoting.

$\underset{\sim}{b}' = (b_i')$ $(i = 1, \ldots, m)$: Solution vector before pivoting. Namely, $\underset{\sim}{b}' \equiv B^{-1} \cdot \underset{\sim}{b}$

$\underset{\sim}{y} = (y_i)$ $(i = 1, \ldots, n)$: Solution vector after pivoting, using r_p^a as the leaving variable. The value is given by:

$$y_i = \begin{cases} b_r'/a_{rq}' & (i = q) \\ b_i' - (a_{iq}' \cdot y_q) & (i \le m, i \ne r) \\ 0 & (i > m, i \ne q, i = r) \end{cases}$$

$X^a \equiv \{x_p^\pi : a \in p, x_p^\pi$ is in the basis before pivoting for $\forall \pi \in \Pi\}$

Namely, X^a is a set of all basic commodity path flow variables that go through arc a before pivoting.

Case 1) When $X^a \neq \varnothing$

All basic variables in X^a become zero after pivoting since the only basic restoration path variable of arc a leaves the basis.

Case 1-a) When $b_r' > 0$,

Since $b_r' > 0$, there exists at least one commodity flow variable in X^a such that it takes a positive value before pivoting. Suppose that x_p^π is such a variable, and it is in the k-th column of the basis ($b_k' > 0$). $a_{rq}' > 0$ since the r-th variable is selected as a leaving variable. Thus, we have $y_q = b_r'/a_{rq}' > 0$. Since all basic variables in X^a becomes zero after pivoting,

$$y_k = b_k' - (a_{kq}' \cdot y_q) = 0$$

This together with the positivity of b_k' and y_q gives

$$a_{kq}' = b_k'/y_q > 0 \qquad (C\text{-}1)$$

Furthermore, we can derive

$$y_q = b_r'/a_{rq}' = b_k'/a_{kq}' \qquad (C\text{-}2)$$

From (C-1) and (C-2), we can conclude that x_p^π is an alternative leaving variable.

Case 1-b) When $b_r' = 0$,

Since $b_r' = 0$, all variables in X^a must be zero even before pivoting. Let I^a be a set of the column indices for all variables in X^a. Assume that there is no other choice for a leaving variable. This implies $a_{kq}' \leq 0$ for $\forall k \in I^a$. Otherwise, a variable violating this condition can leave the basis since

$$y_q = b_r'/a_{rq}' = b_k'/a_{kq}' = 0 \text{ for } k \in I^a$$

Now consider the following linear system which is composed of all basic variables and the entering variable (the q-th column).

$$\begin{array}{lll} x_1 & +a_{1q}'x_q = b_1' \\ x_2 & +a_{2q}'x_q = b_2' \\ \ddots & \vdots \quad \vdots \\ x_m + a_{mq}'x_q = b_m' \end{array} \qquad \textit{(C-3)}$$

The full restorability constraint for arc a can be rewritten in terms of x_i as follows:

$$x_r - \sum_{k \in I^a} x_k - \psi_a \cdot x_q = 0 \qquad \textit{(C-4)}$$

$$\psi_a = \begin{cases} 1 & \text{if the entering column is the commodity flow variable} \\ & \text{and its path contains arc } a \\ 0 & \text{otherwise} \end{cases}$$

Then, any solution of the linear system (C-3) must satisfy (C-4), since the former is the system transformed from the set of the constraints of the SCFA-MF-LINE problem, including Equation (C-4). Now let $x_q = \delta > 0$. Since $a_{rq}' > 0$ and $a_{kq}' \le 0$ for $\forall k \in I^a$,

$x_r = b_r' - a_{rq}'x_q = -a_{rq}'\delta < 0$, and

$x_k = b_k' - a_{kq}'x_q = -a_{kq}'\delta \ge 0$ for $\forall k \in I^a$

However, this violates Equation (C-4) and leads to a contradiction. Therefore, there is at least one variable in X^a with column index k ($k \in I_a$) such that $a_{kq}' > 0$, and we can select this variable as a leaving variable.

Case 2) When $X^a = \varnothing$

The only restoration flow variable of arc a in the basis becomes non-basic. Thus, the entering variable must be a commodity flow variable whose path includes arc a. Otherwise, the row vector of the basis matrix corresponding to the full restorability constraint of arc a becomes zero. Thus, the resulting column vectors of the basis matrix do not form the basis. Since all restoration flow variables for arc a are out of the basis after pivoting, $y_q = 0$. Thus, $b_r' = y_q \cdot a_{rq}' = 0$. Substitute $x_q = \delta > 0$ in the system (C-3) as before. This yields $x_r = b_r' - a_{rq}'x_q = -a_{rq}'\delta < 0$. However, this violates Equation (C-4). Therefore, the only basic restoration flow variable of arc a never leaves the basis when $X^a = \varnothing$.

(Q.E.D.)

An initial basis includes one restoration path variable per arc. Thus, the situation described in Lemma C-2 is the only time when all the restoration path variables of a certain arc become non-basic. The Lemma assures that the selected leaving variable can be replaced in such a case. Note that this situation seldom happens in practice, as discussed earlier, and has never been encountered in our experiments. The following concluding theorem on a basis matrix arrangement is a direct consequence of Lemma C-1 and Lemma C-2.

Theorem C-1

Assume $q^{\pi} > 0$ for $\forall \pi \in \Pi$. Then, at each iteration of the sub-process, it is possible to maintain a *standard* form of the basis matrix B by selectively choosing a leaving variable.

The following Lemma ensures that a well-known shortest path algorithm can be applied in the dual feasibility tests related to the commodity flow variables (Lemma C-3-i) as well as the restoration flow variables (Lemma C-3-ii).

Lemma C-3

At the end of the sub-process, the following two statements are true:

i) No negative arc exists in the network G with arc cost $\{d_a + w_a\}$.

ii) For $\forall l \in E$, no negative arc exists in the network $G \backslash \{l\}$ with arc cost $\{\mu_a^l\}$.

(proof)

Obviously $\mu_l^a \geq 0$ $(\forall a \in A, \forall l \in E_a)$ at the end of the subprocess because all dual feasibility conditions are assured except for those related to non-generated flow variables. Let $(\tilde{\sigma}, \tilde{w}, \tilde{\mu})$ denote a row vector of simplex multipliers arranged in the order of corresponding constraints in the standard form of the current basis matrix, B. Then, the equality $(\tilde{\mathbf{d}}_1, \mathbf{0}, \tilde{\mathbf{d}}_3) = (\tilde{\sigma}, \tilde{w}, \tilde{\mu}) B$ holds, where $\tilde{\mathbf{d}}_1$ and $\tilde{\mathbf{d}}_3$ are the cost vectors of the corresponding basic variables. The second equation, $\mathbf{0} = \tilde{w} \cdot B$, is equivalent to $\tilde{w} = -\tilde{\mu} H$ $(H \equiv (H_1, H_2, H_3)^t)$, and this leads to $w_a = \sum_{b \in p} \mu_b^{l \ni a} \geq 0$ for $\forall a \in A$, where p is the key restoration path for arc a. Thus, $d_a + w_a \geq 0$ for $\forall a \in A$.

Appendix D. Extension to SCFA-KSP-LINE

A heuristic algorithm is developed to find a fully restorable capacity assignment for a KSP-based system. It uses the optimal solution of the SCFA-MF-LINE problem as a starting point. The KSP restoration usually requires slightly more spare bandwidth than the MF restoration. The proposed heuristic detects the amount of unrecoverable flow and gradually adds capacity over the shortest restoration path by iteratively invoking the following loop until full restorability is attained:

1. Let u^{α} and u^{β} be the amount of unrecoverable flow of arcs α and β against a failure of link l ($l = \{\alpha, \beta\}$). Calculate $\Delta c_a = \max_{l \in L} \{\varepsilon \cdot \hat{z}_a^l\}$ for $\varepsilon \in (0, 1]$ where $\hat{z}_a^l$ is given by:

   $$\hat{z}_a^l = \theta_{a, p^{\alpha}} \cdot u^{\alpha} + \theta_{a, p^{\beta}} \cdot u^{\beta},$$

   p^{α} and p^{β} are the shortest restoration path for arcs α and β, respectively. $\hat{z}_a^l$ gives the necessary additional capacity of arc a in order to restore the unrecoverable flow due to a failure of link l over the shortest restoration route.
2. For $\forall a \in A$, add capacity by Δc_a.
3. Invoke the round-up procedure.

Fully restorable capacity assignment is obtained in one iteration if the incremental rate, ε, is set to one. However, the additional spare capacity over the shortest restoration routes for certain arcs could increase the spare capacity of the k-th ($k \neq 1$) shortest route of other arcs. On such occasions, the spare capacity can be shared with different failure scenarios, and the required amount of additional spare capacity would decrease. If ε is set to one, however, the heuristic cannot attain the cost reduction by spare capacity sharing. Thus, a smaller value of ε would yield a less expensive assignment with a slightly higher number of iterations. Our preliminary experiments have shown that ε=0.1 is small enough to acquire the least expensive capacity allocation with this heuristic algorithm. This post process routine typically terminates in tens of seconds with a few iterations.

The KSP rerouting performs restoration path hunting through message flooding. In order to complete the restoration in a short period, it does not usually seek lengthy alternate routes, while such routes could be selected in the MF rerouting algorithm. In addition, a hop-limit is typically imposed on a restoration route. Therefore, the solution of the SCFA-MF-LINE problem might not be appropriate as a starting point of the proposed post-processing routine. In order to conform to more realistic situations, the algorithm is slightly modified as follows: First, all candidate restora-

tion routes are enumerated, and their columns are generated at the beginning of the procedure. In the master process, all procedures related to restoration flow variables are skipped, and only the dual feasibility conditions corresponding to commodity flow variables (namely, Equation (A-4)) are examined. If necessary, a commodity flow variable is generated in the master process, but not a restoration flow variable. The post-processing routine uses the solution of this revised algorithm as its initial solution. Due to the restriction on the restoration path variables, significant improvement on the computation time is achieved by this modification. Our experiment indicates that the required CPU time is reduced by an order of magnitude for a US-WAN network. It is also observed that the cost increase is only 2 to 10 percent over MF-based networks with restoration path restriction.

Appendix E. SCFA-MF-ETE: Problem formulation

The SCFA-MF-ETE searches for an optimal capacity and flow assignment for the self-healing ATM networks based on end-to-end restoration. In the problem formulation, the full restorability constraints must be revised due to the different restoration strategies, while the others remain almost identical to that of the SCFA-MF-LINE problem. In end-to-end restoration, each commodity selects its rerouting paths independently between the respective source and destination nodes. Thus, one full restorability constraint is required for every combination of a commodity π and a failed link l, which is referred to as a (π,l) *constraint* or a (π,l) *row* in this project. The following additional notations are employed in the formulation of the SCFA-MF-ETE problem[11]:

$RP^{\pi, l}$: A set of all possible restoration routes for commodity $\pi \in \Pi$ against a failure of link $l \in E$. Its element is called a (π,l) *restoration path.*

$\underset{\sim}{r} = (r_p^{\pi, l})$: A restoration path flow vector. $\pi \in \Pi$, $l \in E$, $p \in RP^{\pi, l}$

$\theta_{a, l, p}$: An indicator variable which equals 1 if both arc a and link l are contained in path p, and 0 otherwise.

11. Refer to Appendix T for a complete listing of notations.

$r_a^l \ \forall a \in A, \forall l \in E_a$: The amount of bandwidth of arc a which is released by affected VP's due to a failure of link l.

$w^{\pi,l} \ \forall \pi \in \Pi, \forall l \in E$: The price of link restoration per commodity (dual variable).

Now the SCFA-MF-ETE problem and its dual problem (SCFA-MF-ETE-DUAL) are formulated as follows:

Minimize $D(\underline{x}, \underline{r}, \underline{z}) = \sum_{a \in A} d_a \cdot c_a$ **(SCFA-MF-ETE)**

over $\underline{x} \geq 0, \underline{r} \geq 0 \ \underline{z} \geq 0$

subject to a) $\sum_{p \in P^\pi} x_p^\pi = q^\pi \qquad \forall \pi \in \Pi$ *(E-1)*

b) $z_a + r_a^l - z_a^l \geq 0 \qquad \forall a \in A, \forall l \in E_a$ *(E-2)*

c) $\sum_{p \in RP^{\pi,l}} r_p^{\pi,l} - \sum_{a \in l} \sum_{p \in P^\pi} \theta_{a,p} \cdot x_p^\pi = 0 \qquad \forall \pi \in \Pi, \forall l \in E$ *(E-3)*

$c_a = f_a + z_a$

$f_a = \sum_{\pi \in \Pi} \sum_{p \in P^\pi} \theta_{a,p} \cdot x_p^\pi$

$z_a^l = \sum_{\pi \in \Pi} \sum_{p \in RP^{\pi,l}} \theta_{a,p} \cdot r_p^{\pi,l}$

$r_a^l = \sum_{\pi \in \Pi} \sum_{p \in P^\pi} \theta_{a,l,p} \cdot x_p^\pi$

Maximize $\sum_{\pi \in \Pi} q^\pi \cdot \sigma^\pi$ **(SCFA-MF-ETE-DUAL)**

over $\sigma^\pi, w^{\pi,l}$ unrestricted, $\mu_a^l \geq 0$

$$\text{subject to i)} \quad \sum_{a \in p:(l \ni a)} (d_a + w^{\pi,l}) - \sum_{l \in p} \sum_{a \in p} \mu_a^l \geq \sigma^\pi \qquad \forall p \in P^\pi, \forall \pi \in \Pi \qquad \textit{(E-4)}$$

$$\text{ii)} \quad \sum_{a \in p} \mu_a^l \geq w^{\pi,l} \qquad \forall p \in RP^{\pi,l}, \forall \pi \in \Pi, \forall l \in E \qquad \textit{(E-5)}$$

$$\text{iii)} \quad \sum_{l \in E_a} \mu_a^l \leq d_a \qquad \forall a \in A \qquad \textit{(E-6)}$$

$$\mu_a^{l:(a \in l)} \equiv 0$$

The constraints (E-1), (E-2) and (E-3) correspond to the flow conservation law, the capacity constraints, and full restorability constraints, respectively. Note that the end-to-end restoration scheme releases the bandwidth of the affected VP's over their original routes. The released capacity can be reclaimed for the restoration of any affected traffic. This phenomenon is embodied in the second term of the capacity constraints which express the amount of the reusable bandwidth. σ^π, μ_a^l and $w^{\pi,l}$ are the simplex multipliers corresponding to the constraints (E-1), (E-2) and (E-3), respectively.

Appendix F. SCFA-MF-ETE: Solution approach

An approach similar to that for the SCFA-MF-LINE problem can be applied to the SCFA-MF-ETE problem. The arc-chain flow representation is employed to simplify the constraint structure, and the column generation approach is applied to handle the enormous number of path variables due to the arc-chain flow representation. The solution procedure is decomposed into a master process and a sub-process. The computational load of the simplex procedure in the sub-process is considerably lowered by exploiting the special structure of the problem. The following two statements are true at the end of any iteration if $q^\pi > 0$ for $\forall \pi \in \Pi$.

1. For each commodity $\pi \in \Pi$, at least one commodity path flow variable, x_p^π ($p \in P^\pi$), is in the basis.
2. For each combination of $\pi \in \Pi$ and $l \in E$, it is possible to keep at least one restoration path flow variable, $r_p^{\pi,l}$ ($p \in RP^{\pi,l}$), in the basis.

Then the basis matrix, B, can be arranged as shown below, and it can be readily factorized into an LU form:

$$B = \begin{bmatrix} I & & & C \\ R & I & & D \\ A_1 & H_1 & -I & K_1 & M_1 \\ A_2 & H_2 & & I & M_2 \\ A_3 & H_3 & & K_3 & M_3 \end{bmatrix} = \begin{bmatrix} I & & & \\ R & I & & \\ A_1 & H_1 & I & \\ A_2 & H_2 & & I \\ A_3 & H_3 & & K_3 & \hat{L} \end{bmatrix} \cdot \begin{bmatrix} I & & & C \\ & I & & F \\ & & -I & K_1 & Y_1 \\ & & & I & Y_2 \\ & & & & \hat{U} \end{bmatrix} \qquad (F\text{-}1)$$

where $F = D - RC$, $Y_i = M_i - H_i \cdot F - A_i \cdot C$ for $i = 1, 2, 3$ and $Z \equiv \hat{L} \cdot \hat{U} = Y_3 - K_3 \cdot Y_2$. The arrangement of the rows and columns is consistent with that in Equation (B-1). The structure is almost identical except for the introduction of A_i (i=1,2,3), which comes from the second term of Equation (E-2). All sub-matrices, R, A_i, H_i, K_i, and M_i (i=1,2,3) are sparse. We again call the basis matrix arranged in this manner a *standard* form. A direct update of the $\hat{L}\hat{U}$ submatrices can be carried out in the same fashion as discussed in the previous section.

Additional mechanisms must be devised to tackle the new issues inherent to the SCFA-MF-ETE problem. First of all, the number of full restorability constraints grows enormously, from $|A|$ ($O(A)$) in the SCFA-MF-LINE problem to $|E| \cdot |\Pi|$ ($O(AV^2)$ in the SCFA-MF-ETE problem (i.e. from 46 to 2,530 for an NJ-LATA network, and from 90 to 34,020 for a US-WAN network). Although the size of the submatrix Z and the computation of $\hat{L}\hat{U}$ factorization remain the same, the revised simplex procedure significantly slows down in the computation of an entering column vector and simplex multipliers at each iteration. The end-to-end restoration scheme requires that its full restorability constraints be expressed for all combinations of commodities and link failure scenarios. However, the (π, l) constraint is necessary only if at least one generated commodity flow for π, x_p^π, goes through the link l. The (π, l) constraint is defined *active* if it satisfies the above condition and is called *inactive* otherwise. The inactive (π, l) constraints do not have any effect on the sub-process since the value of their related basic variables, $r_p^{\pi, l}$, stays zero in the entire operation of the sub-process. Since each commodity is expected to be transferred over only a fraction of a network, the number of the constraints can be largely reduced by generating only active (π, l) constraints.

The master process generates a (π, l) row when it becomes active due to newly generated commodity flow variables of π. In addition, it deletes non-basic commodity and restoration flow vari-

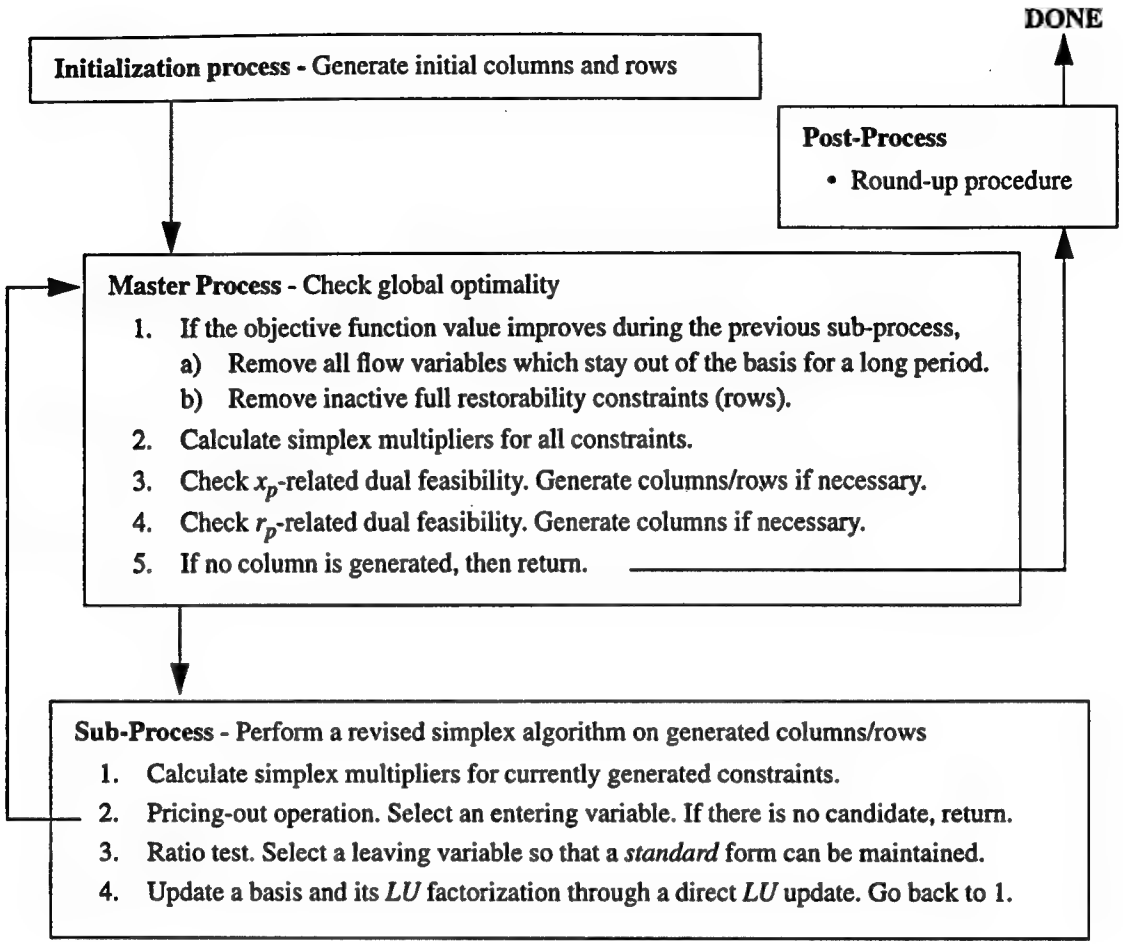


Figure F-1. Solution procedure for the SCFA-MF-ETE problem

ables which have not been referred to for some period. This decreases the number of non-basic variables involved in the pricing-out operation [54] in the subprocess. More importantly, the dimension of the basis matrix can be reduced. A (π, l) row can be removed if it turns inactive as a result of the deletion of a non-basic commodity flow variable. Note that this column deletion procedure can be performed only when the value of the objective function has improved in the previous sub-process. This condition is required to guarantee the termination of the algorithm in a finite number of iterations. Figure F-1 summarizes the proposed solution procedure. The initialization process generates only the shortest commodity path variables, active full restorability constraints, and the shortest restoration path variables for the active constraints.

The major role of the master process is to test the dual feasibility conditions and to determine

whether the solution obtained in the sub-process is globally optimal or not. Since the dual feasibility condition (E-6) and non-negativity of μ_a^l are already established at the end of the sub-process, only the conditions (E-4) and (E-5) must be checked. As in the SCFA-MF-LINE problem, it is sufficient to examine a set of shortest paths to verify the global optimality. However, a shortest path algorithm cannot be applied to the test on (E-4). This condition is assured if the length of the shortest path of commodity π with the modified arc cost $(d_a + w^{\pi, l \ni a})$ and the mutual arc cost $(-\mu_a^l)$ is not less than σ^π. The mutual arc cost is imposed if a path goes through both arc a and link l. A quadratic shortest path (QSP) algorithm, which is described in Appendix Q, is developed to obtain the shortest path of this type. As to the condition (E-5), a traditional shortest path algorithm can be adopted. This condition is satisfied if the length of the shortest path $p \in RP^{\pi, l}$ under modified arc length μ_a^l in the network $G \backslash \{l\}$ is not less than $w^{\pi, l}$. If the dual feasibility conditions are violated, then the master process generates all violated shortest paths and invokes the sub-process. Otherwise, the current solution is globally optimal.

In the above dual feasibility test, all simplex multipliers must be at hand including the ones corresponding to non-generated full restorability constraints. The multiplier for the generated rows can be obtained by $\tilde{\mathbf{d}} \cdot B^{-1}$ where a vector $\tilde{\mathbf{d}}$ contains the cost vector corresponding to the current basis. As for a non-generated (π, l) constraint, its corresponding simplex multiplier, $w^{\pi, l}$, can be obtained by $\sum_{a \in p} \mu_a^l$, where p is the shortest path for commodity π in the network $G \backslash \{l\}$ with the modified arc cost μ_a^l. It can also be shown that the row generation and deletion procedures in the master process will not change the value $Z \equiv \hat{L} \cdot \hat{U}$ in Equation (F-1). Therefore, it is not necessary to re-factorize Z at the end of the master process, and the LU factorization for a new set of constraints can be trivially obtained.

The execution time of the proposed algorithm is largely reduced by the above mechanisms, although more time is required than in the case of the SCFA-MF-LINE problem. In our experiment[12], the execution time ranges from an hour to a day depending on network size. Since the algorithm is meant for a network planning, the execution time is not a critical issue unless it takes unreasonably long, say a month or more.

12. Refer to Section 3.4.2. for details on the settings of the experiments.

Appendix G. SCFA-MF-ETE: Validity of the algorithm

This section shows the validity of the claims which have been employed in the development of our algorithm. First of all, we show that a basis matrix can be arranged in a *standard* form at any iteration of the sub-process.

Lemma G-1

Consider an active (π, l) constraint. Suppose that $r_p^{\pi, l}$ is the only (π, l) restoration path variable in the basis and is selected as a leaving variable at some point in the sub-process. Furthermore, suppose that the entering variable is not a (π, l) restoration path variable. Then, we can always find a candidate for a leaving variable other than $r_p^{\pi, l}$.

(proof)

Let $X^{\pi, l} \equiv \{ x_p^{\pi} : l \in p, x_p^{\pi}$ is in the basis before pivoting $\}$ Then, the assertion can be proved in the same way as in the proof of Lemma C-2 with X^a replaced by $X^{\pi, l}$. Namely, it can be shown that some commodity flow variable in $X^{\pi, l}$ can leave the basis when $X^{\pi, l}$ is non-empty, and that $r_p^{\pi, l}$ is not selected as a leaving variable when $X^{\pi, l}$ is empty.

(Q.E.D.)

An initial basis includes one restoration path variable per active restoration path. When a (π, l) row is generated in the master process, a corresponding restoration path variable is also generated. Thus, the situation described in Lemma G-1 is the only occasion when all (π, l) restoration path variables could be removed from the basis. As in the case of the SCFA-MF-LINE problem, the above Lemma assures that replacing a leaving variable is always possible on such an occasion. Unlike the previous case, however, this situation could happen in practice[13]. Therefore, the selection mechanism of a leaving variable is mandatory for the SCFA-MF-ETE solution module.

Theorem G-2

Assume $q^{\pi} > 0$ for $\forall \pi \in \Pi$. Then, at the end of an iteration, a new basis matrix can be

13. This situation could happen if more than one commodity flow variables of π are generated and if the demand of π is carried over only a part of the generated paths. If link l is contained only in the generated commodity paths with no flow, then the only (π, l) restoration path variable in the basis can be selected as a leaving variable. Even though it is a relatively rare event, the situation has occasionally appeared in our experiments.

arranged in the *standard* form given in Equation (F-1).

(proof)

Since $q^{\pi} > 0$, there is at least one basic commodity flow variable for each commodity. Thus it is possible to choose a key column per commodity and to arrange the first set of columns. Lemma G-1 ensures that at least one key restoration flow column can be kept in the basis. Thus, it is possible to arrange the second set of columns. The third and fifth sets of columns can be trivially arranged, while the arrangement of the fourth set of columns can be proved by the same approach as in Lemma C-1.

(Q.E.D.)

The following three Lemmas deal with the validity of the procedures employed in the master process. Lemma G-3 proves that a shortest path algorithm and the proposed QSP algorithm can be applied in the dual feasibility test. Lemma G-4 assures the invariability of the $\hat{L}\hat{U}$ sub-matrices after the row generation and deletion procedure. Finally, Lemma G-5 tells how to determine the simplex multipliers corresponding to inactive (π, l) constraints.

Lemma G-3

At the end of the sub-process, the following two inequalities hold[14].

$$d_a + w^{\pi, l \ni a} - \sum_{l' \in E_a} \mu_a^{l'} \geq 0 \text{ for } \forall a \in A$$

$$\mu_a^{l} \geq 0 \text{ for } \forall a \in A, \forall l \in E_a$$

(proof)

Obviously, the dual feasibility conditions, $\mu_a^l \geq 0$ and $d_a - \sum_{l' \in E_a} \mu_a^{l'} \geq 0$ (Equation (E-6)), hold for $\forall a \in A$ and $\forall l \in E_a$ at the end of the sub-process. Thus, it suffices to show $w^{\pi, l} \geq 0$ to prove the Lemma. The same argument as in Lemma C-2 can be applied to show $w^{\pi, l} = \sum_{a \in p} \mu_a^l \geq 0$ for all active (π, l) rows, where p is the key restoration path for commodity π against a failure of link l. For non-generated (π, l) rows, $w^{\pi, l}$ is selected as $\sum_{a \in p} \mu_a^l \geq 0$, where p is the shortest (π, l) restoration path with modified arc cost $\{\mu_a^l\}$.

(Q.E.D.)

14. The first condition is necessary for our proposed QSP algorithm to function correctly. See Appendix Q for details.

Lemma G-4

The row generation and deletion procedures in the master process do not change the value of $Z \equiv \hat{L} \cdot \hat{U}$ in Equation (F-1).

(proof)

Suppose that the (π, l) row and its related key restoration flow variable, $r_p^{\pi, l}$, are generated in the master process. $r_p^{\pi, l}$ is the only basic column vector with a non-zero entry in the (π, l) row after the row generation. Therefore, no column vector of $F^{new} = D^{new} - R^{new} C$ has a non-zero entry at the (π, l) row, where the superscript, *new*, in a sub-matrix indicates the sub-matrix after the row generation.. Thus,

$$Y_i^{new} = M_i - H_i^{new} \cdot F^{new} - A_i \cdot C = M_i - H_i^{old} \cdot F^{old} - A_i \cdot C = Y_i^{old}$$

where the superscript, *old*, in a sub-matrix indicates the sub-matrix before the row generation. Since there is no change in the sub-matrix K_3,

$$Z^{new} = Y_3^{new} - K_3 \cdot Y_2^{new} = Y_3^{old} - K_3 \cdot Y_2^{old} = Z^{old}.$$

Now suppose that the (π, l) row and its related key restoration flow variable, $r_p^{\pi, l}$, are removed in the master process. Again, $r_p^{\pi, l}$ is the only basic column vector with non-zero entry in the (π, l) row before the row deletion, and the same argument as above can be applied to derive $Z^{new} = Z^{old}$.

(Q.E.D.)

Lemma G-5

Suppose that the (π, l) row is not generated. Let p be the shortest restoration path for commodity π against a failure of link l in the network with the arc cost $\{\mu_a^l\}$. Then, the dual feasibility condition (E-5) is guaranteed for all (π, l) restoration path variables if the simplex multiplier $w^{\pi, l}$ is selected as $w^{\pi, l} = \sum_{a \in p} \mu_a^l$.

(proof)

Due to the structure of the column vector of the restoration variable, the simplex multiplier is given by $w^{\pi, l} = \sum_{a \in p} \mu_a^l$ for a certain path, $p \in RP^{\pi, l}$. In addition, the value of μ_a^l is determined by the basis matrix B and independent of $w^{\pi, l}$. Thus, in order to satisfy the dual

feasibility condition (E-5) for all (π, l) restoration path variables, p must be the shortest path in $G \setminus \{l\}$ with the arc cost $\{\mu_a^l\}$.

(Q.E.D.)

Finally, the following two theorems assure the validity of the proposed algorithm.

Theorem G-6

The proposed algorithm terminates in a finite number of iterations.

(proof)

Let Q be a set of all commodity flow columns and restoration flow columns, and let Q_k be the subset of Q containing all generated columns upon the k-th invocation of the sub-process. Since the sub-process is guaranteed to produce the optimal value with the currently generated columns, Q_{k+1} is different from Q_k if the objective function value improves at the k-th iteration. Since the objective function value will never increase, the subset Q_k will not appear in a later iteration. If there is no improvement, then the subset keeps growing, and thus it does not return to a previous state. Since the number of subsets of Q is finite, the algorithm terminates in a finite number of iterations.

(Q.E.D.)

Theorem G-7

The proposed algorithm terminates with a globally optimal solution.

(proof)

When the algorithm terminates, dual feasibility is assured for all generated rows. Furthermore, Lemma G-5 guarantees dual feasibility for all non-generated rows. Since the solution is also primally feasible, it is globally optimal [53].

(Q.E.D.)

Appendix H. SVPR-MF-LINE: Problem formulation

The SVPR problem shares a common structure with the SCFA problem. A major difference resides in the constraint c) where full restorability constraints are replaced by restoration flow constraints. The restoration flow constraints refer to restoring the affected flow over restoration paths. If there is not enough spare bandwidth, a part of the affected bandwidth could be lost. In order to accommodate this phenomenon, a lost flow variable $\underset{\sim}{t} = (t_a)$ $(a \in A)$ is introduced in the formulation of the SVPR problem. t_a takes the amount of an unrecoverable flow of arc a due to a failure of link l $(l \ni a)$.

Now, the SVPR-MF-LINE problem is formulated as a large-scale linear programming as in the SCFA-MF-LINE problem:

Given $G = (V, A, \underset{\sim}{c})$ and Q, **(SVPR-MF-LINE)**

Minimize $L \cdot |E| = \sum_{a \in A} t_a$

over $\underset{\sim}{x} \geq 0$, $\underset{\sim}{r} \geq 0$, and $\underset{\sim}{t} \geq 0$

subject to a) $\sum_{p \in P^\pi} x_p^\pi = q^\pi \quad \forall \pi \in \Pi$ *(H-1)*

b) $f_a + z_a^l \leq c_a \quad \forall a \in A, \forall l \in E_a$ *(H-2)*

c) $-f_a + \sum_{p \in RP^a} r_p^a + t_a = 0 \quad \forall a \in A$ *(H-3)*

where $f_a = \sum_{\pi \in \Pi} \sum_{p \in P^\pi} \theta_{a,p} \cdot x_p^\pi$ and $z_a^l = \sum_{b \in l} \sum_{p \in RP^b} \theta_{a,p} \cdot r_p^b$

The arc-chain flow representation is employed. The constraints (H-1), (H-2) and (H-3) correspond to the flow conservation law, the capacity constraints and the restoration flow constraints, respectively. In the SVPR problem, z_a^l expresses the amount of total restoration flow over arc a upon a failure of link l. The restoration flow constraints state that the amount of an unrecoverable flow of arc a plus the sum of restoration flows for the arc is equal to the arc flow (the amount of a

flow to be recovered). The dual formulation of the above problem is given by (SVPR-MF-LINE-Dual).

Maximize $\sum_{\pi \in \Pi} q^{\pi} \cdot \sigma^{\pi} + \sum_{a \in A} c_a \cdot \left(\sum_{l \in E_a} \mu_a^l \right)$ **(SVPR-MF-LINE-Dual)**

over σ^{π}, w_a unrestricted, $\mu_a^l \leq 0$

subject to i) $\sum_{a \in p} \left(w_a - \sum_{l \in E_a} \mu_a^l \right) \geq \sigma^{\pi} \quad \forall p \in P^{\pi}, \forall \pi \in \Pi$ *(H-4)*

ii) $\sum_{b \in p} (-\mu_b^{l \ni a}) \geq w_a \quad \forall p \in RP^a, \forall a \in A$ *(H-5)*

iii) $w_a \leq 1 \quad \forall a \in A$

The dual variables σ^{π}, μ_a^l and w_a are the simplex multipliers corresponding to the constraints (H-1), (H-2) and (H-3), respectively.

Appendix I. SVPR-MF-LINE: Solution approach

Due to the similarity in their formulations, the strategies developed for the SCFA-MF-LINE problem can be applied to the SVPR-MF-LINE problem with a slight modification. The arc-chain flow representation and the column generation approach are employed to reduce the size of constraints. The solution process is decomposed into the master process and the subprocess. Only a part of commodity path flow variables x_p^{π} and restoration path flow variables r_p^a are generated. The master process tests the global optimality by checking the conditions (H-4) and (H-5). A shortest path algorithm can be applied for this test. In the subprocess, *LU* decomposition of the basis matrix is calculated by exploiting the special structure of the matrix and is manipulated through the direct *LU* update mechanism introduced in Section 3.2.2.. The following two statements can be shown to be true at the end of all iterations of the subprocess if $q^{\pi} > 0$ for $\forall \pi \in \Pi$:

1. For each commodity $\pi \in \Pi$, at least one commodity path flow variable, x_p^{π} ($p \in P^{\pi}$), is in

the basis.

2. For each arc $a \in A$, it is possible to maintain at least one restoration path flow variable, r_p^a ($p \in RP^a$), or the corresponding lost flow variable, t_a, in the basis.

Using this observation, the basis matrix B can be arranged in the following *standard* form at each iteration, which can be readily decomposed into an *LU* form:

$$B = \begin{bmatrix} I & & C \\ R & I & D \\ A_1 & H_1 & I & M_1 \\ A_2 & H_2 & & M_2 \end{bmatrix} = \begin{bmatrix} I & & \\ R & I & \\ A_1 & H_1 & I & \\ A_2 & H_2 & & \hat{L} \end{bmatrix} \cdot \begin{bmatrix} I & & C \\ & I & F \\ & & I & Y_1 \\ & & & \hat{U} \end{bmatrix} \qquad (I\text{-}1)$$

where $F = D - RC$, $Y_i = M_i - A_iC - H_iF$ for $i = 1, 2$ and $Z \equiv \hat{L} \cdot \hat{U} = Y_2$. The first set of columns corresponds to the key commodity path flow variables, and the second set contains the key restoration path flow variables or lost flow variables. The remaining commodity and restoration path flow variables are collected into the last set, while the slack variables for the capacity constraints are arranged in the third set of columns. Note that all lost flow variables in the basis are placed in the second set. All basic restoration path variables for arc a are arranged into the last set if the corresponding lost flow variable t_a is in the basis. The first set of rows corresponds to the flow conservation laws, the second to the restoration flow constraints, and the last two represent capacity constraints.

Figure I-1 illustrates the structure of the solution procedure. The structure is almost identical to that for the SCFA-MF-LINE problem. A major difference resides in the initialization process. Unlike the SCFA-MF-LINE problem, there is no obvious feasible solution satisfying the capacity constraints. This issue can be circumvented by the two-phase approach of the simplex algorithm with the introduction of artificial variables [54]. Details are given in Appendix R.1. In brief, the initialization process takes the following steps:

1. Obtain a shortest route commodity flow. If it is feasible, go to Step 3.
2. Find an initial feasible solution by solving an LP problem with artificial variables.
3. Set all restoration flow variables as non-basic.
4. Calculate the amount of lost flow per each arc. Set all lost flow variables as basic and place them in the second set of columns in Equation (I-1).

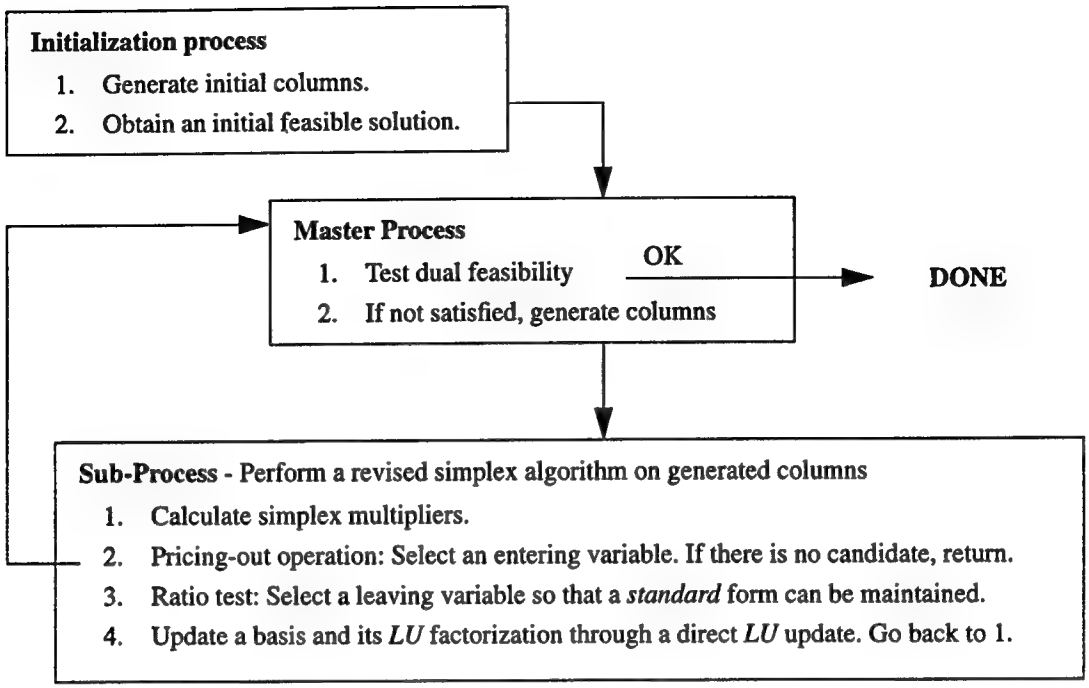


Figure I-1. Solution procedure for the SVPR-MF-LINE problem

Appendix J. SVPR-MF-LINE: Validity of the algorithm

The following Lemmas and Theorems assure the correctness of the proposed algorithm described in the previous section. Since they can be proven based on the same arguments as those in their counterpart in the SCFA-MF-LINE problem, the proof is omitted here.

Lemma J-1

Suppose that the only arc restoration flow or arc lost flow variable for arc a is selected as a candidate for a leaving variable in the sub-process. Furthermore, suppose that an entering variable is not an arc restoration flow or arc lost flow variable for arc a. Then, another candidate can be always found for a leaving variable.

Theorem J-2

Assume $q^{\pi} > 0$ for $\forall \pi \in \Pi$. Then, at each iteration of the sub-process, it is possible to select a leaving variable so that the *standard* form of the basis matrix B can be maintained.

Lemma J-3

At the end of sub-process, the following two statements are true:

i) No negative cycle exists in the network G with arc cost $\{w_a - \sum_{l \in E_a} \mu_a^l\}$.

ii) For $\forall l \in E$, no negative arc exists in the network $G \backslash \{l\}$ with arc cost $\{-\mu_a^l\}$.

Appendix K. SVPR-MF-ETE: Problem formulation

The end-to-end restoration scheme recovers failed virtual paths between their source and destination nodes. Thus, a lost flow variable is defined for each combination of a commodity and a failure event. Let $t^{\pi, l}$ $(\pi \in \Pi, l \in E)$ denote a lost flow variable which takes the amount of an unrecoverable flow of commodity π upon a failure of link l, and define a lost flow vector $\underline{t} = (t^{\pi, l})$. Then, the SVPR-MF-ETE problem is formulated as follows:

Given $G = (V, A, \underline{c})$ and Q, **(SVPR-MF-ETE)**

Minimize $L \cdot |E| = \sum_{l \in E} \sum_{\pi \in \Pi} t^{\pi, l}$

over $\underline{x} \geq 0$, $\underline{r} \geq 0$ and $\underline{t} \geq 0$

subject to a) $\sum_{p \in P^{\pi}} x_p^{\pi} = q^{\pi}$ $\forall \pi \in \Pi$ *(K-1)*

b1) $f_a - r_a^l + z_a^l \leq c_a$ $\forall a \in A, \forall l \in E_a$ *(K-2)*

b2) $f_a \leq c_a$ $\forall a \in A$ *(K-3)*

c) $$-\sum_{a \in l}\sum_{p \in P^{\pi}} \theta_{a,p} \cdot x_p^{\pi} + \sum_{p \in RP^{\pi,l}} r_p^{\pi,l} + t^{\pi,l} = 0 \quad \forall \pi \in \Pi, \forall l \in E \qquad (K\text{-}4)$$

where $$f_a = \sum_{\pi \in \Pi}\sum_{p \in P^{\pi}} \theta_{a,p} \cdot x_p^{\pi}$$

$$z_a^l = \sum_{\pi \in \Pi}\sum_{p \in RP^{\pi,l}} \theta_{a,p} \cdot r_p^{\pi,l}$$

$$r_a^l = \sum_{\pi \in \Pi}\sum_{p \in P^{\pi}} \theta_{a,l,p} \cdot x_p^{\pi}$$

The flow conservation law and the restoration flow constraints are expressed by Equations (K-1) and (K-4), respectively. Unlike the previous formulations, two types of capacity constraints are required: one in the event of a failure (Equation (K-2)) and the other under normal conditions (Equation (K-3)). Since a restoration flow is additionally sent over the existing flow in the case of line-restoration, the capacity constraints in the event of a failure ensure the feasibility of a flow under normal conditions. This is not true for end-to-end restoration since a part of working capacity could be released at the time of a failure. If the amount of the released bandwidth at arc a (r_a^l) is more than the amount of the total restoration flow over the arc (z_a^l) upon a failure of link l, then Equation (K-2) does not imply Equation (K-3). Thus, Equation (K-3) is necessary. Corresponding to this new constraint, an additional simplex multiplier must be introduced in the dual formulation. Define μ_a as the dual variable corresponding to the constraint (K-3), and let $\mu_a^{l \ni a} \equiv \mu_a$. Now the dual formulation of the above problem is obtained as follows:

Maximize $$\sum_{\pi \in \Pi} q^{\pi} \cdot \sigma^{\pi} + \sum_{a \in A} c_a \cdot \left(\sum_{l \in E} \mu_a^l \right)$$ **(SVPR-MF-ETE-Dual)**

over σ^{π}, $w^{\pi,l}$ unrestricted, $\mu_a^l \le 0$

subject to i) $$\sum_{a \in p} \left(w^{\pi, l \ni a} + \sum_{l \in E} (-\mu_a^l) - \sum_{\substack{l \in p \\ a \notin l}} (-\mu_a^l) \right) \ge \sigma^{\pi} \quad \forall p \in P^{\pi}, \forall \pi \in \Pi \qquad (K\text{-}5)$$

ii) $$\sum_{a \in p} (-\mu_a^l) \ge w^{\pi,l} \qquad \forall p \in RP^{\pi,l}, \forall \pi \in \Pi, \forall l \in E \qquad (K\text{-}6)$$

iii) $w^{\pi,l} \leq 1$ $\forall \pi \in \Pi, \forall l \in E$

Appendix L. SVPR-MF-ETE: Solution approach

The strategies developed to solve the SCFA-MF-ETE are applicable to the SVPR-MF-ETE problem due to the common structures of the two problems: the arc-chain flow representation, the column generation and deletion technique, the *LU* decomposition based on a special arrangement of the basic variables, the direct *LU* update, and the row generation and deletion technique. The *standard* basis matrix arrangement and its *LU* decomposition are given by:

$$B = \begin{bmatrix} I & & C \\ R & I & D \\ A_1 & H_1 & I & M_1 \\ A_2 & H_2 & & M_2 \end{bmatrix} = \begin{bmatrix} I & & \\ R & I & \\ A_1 & H_1 & I & \\ A_2 & H_2 & & \hat{L} \end{bmatrix} \cdot \begin{bmatrix} I & & C \\ & I & F \\ & & I & Y_1 \\ & & & \hat{U} \end{bmatrix} \qquad (L\text{-}1)$$

where $F = D - RC$, $Y_i = M_i - A_i C - H_i F$ for $i = 1, 2$ and $Z \equiv \hat{L} \cdot \hat{U} = Y_2$. The structure of the basis matrix is equivalent to that for the SVPR-MF-LINE problem, although each submatrix has different components. Furthermore, the size of the second row and column grows explosively in the SVPR-MF-ETE problem. The following observation allows the basis matrix to be arranged in a standard form. Assume $q^{\pi} > 0$ for $\forall \pi \in \Pi$. Then at the end of any iteration of the subprocess,

1. at least one commodity path flow variable, x_p^{π} ($p \in P^{\pi}$), is in the basis for each commodity $\pi \in \Pi$, and
2. it is possible to maintain at least one restoration path flow variable, $r_p^{\pi,l}$ ($p \in RP^{\pi,l}$), or the corresponding lost flow variable, $t^{\pi,l}$, in the basis, for each combination of $\pi \in \Pi$ and $l \in E$.

Figure L-1 depicts the structure of the solution procedure for the SVPR-MF-ETE problem. In the master process, a dual feasibility test is conducted through a shortest path algorithm (Equation (K-5)) or the quadratic shortest path algorithm (Equation (K-6)) described in Appendix Q. An initial feasible solution can be obtained in the same manner as in the SVPR-MF-LINE problem. Refer to Appendix R.2 for details.

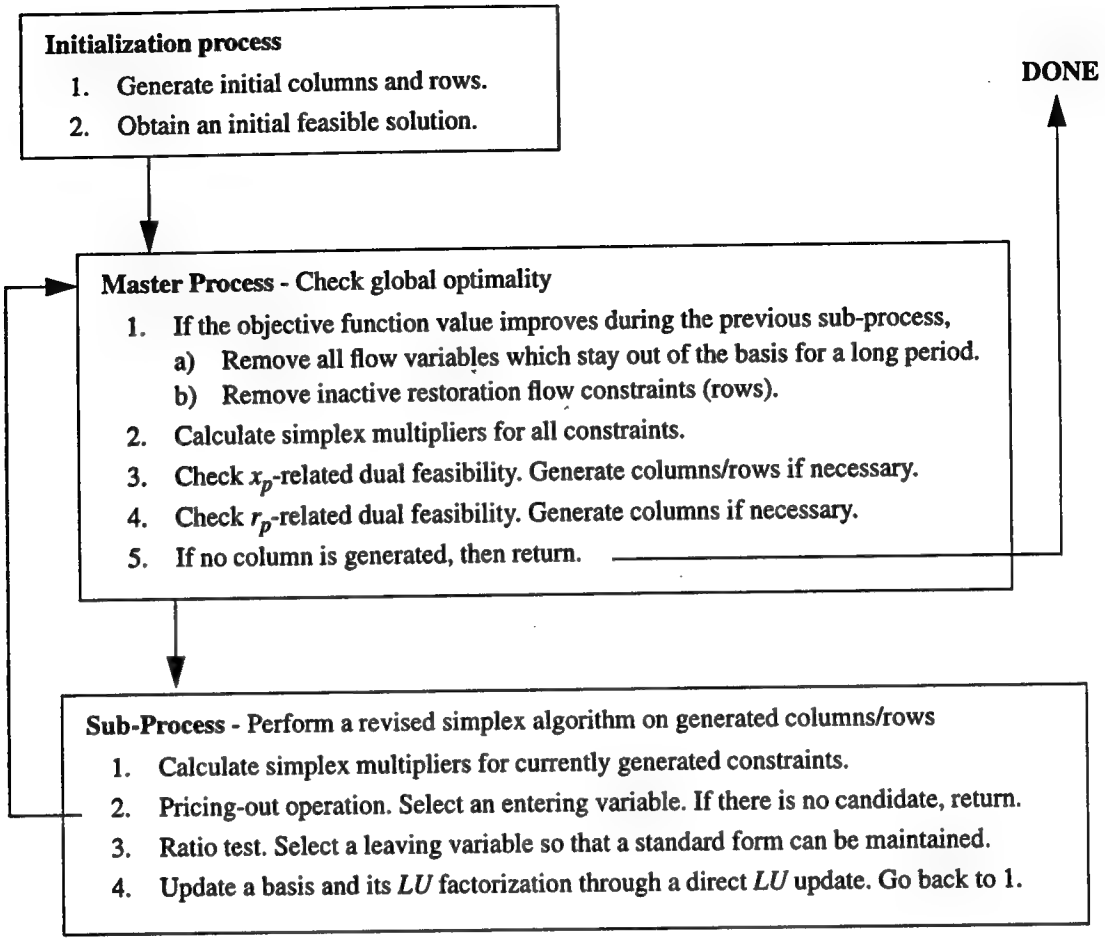


Figure L-1. Solution procedure for the SVPR-MF-ETE problem

Appendix M. SVPR-MF-ETE: Validity of the algorithm

The arguments developed for the SCFA-MF-ETE problems as well as the SVPR-MF-LINE problem can also be applied to verify the proposed algorithm. This section lists related Lemmas and Theorems for the sake of completeness.

Lemma M-1

Consider an active (π, l) constraint. Suppose that the only (π, l) related variable (restoration path variable or lost flow variable) in the basis is selected as a leaving variable at some point in the sub-process. Furthermore, suppose that an entering variable is not a (π, l) related variable. Then, we can always find another candidate for a leaving variable.

Theorem M-2

Assume $q^{\pi} > 0$ for $\forall \pi \in \Pi$. Then, at the end of an iteration a new basis matrix can be arranged in the *standard* form given in Equation (L-1).

Lemma M-3

At the end of the sub-process, the following two inequalities hold.

$$w^{\pi, l \ni a} - \mu_a^{l \ni a} \geq 0 \text{ for } \forall a \in A$$

$$\mu_a^l \leq 0 \text{ for } \forall a \in A, \forall l \in E_a$$

Lemma M-4

The row generation and deletion procedures in the master process do not change the value of $Z \equiv \hat{L} \cdot \hat{U}$ in Equation (L-1).

Lemma M-5

Suppose that the (π, l) row is not generated. Let p be the shortest restoration path for commodity π against a failure of link l in the network with the arc cost $\{-\mu_a^l\}$. Then, the dual feasibility condition (K-6) is guaranteed for all (π, l) restoration path variables if the simplex multiplier $w^{\pi, l}$ is selected as $w^{\pi, l} = -\sum_{a \in p} \mu_a^l$.

Theorem M-6

The proposed algorithm terminates in a finite number of iterations.

Theorem M-7

The proposed algorithm terminates with a globally optimal solution.

Appendix N. SVPR-KSP-LINE: Problem formulation

Different problem formulations can lead to different solution procedures. We consider two formulations which differ in the way they accommodate restoration flow constraints. The first formulation (SVPR-KSP-LINE1) embeds such constraints into an objective function.

Given $G = (V, A, \underline{c})$ and Q, **(SVPR-KSP-LINE1)**

Minimize $$L(\underline{f}) = \frac{1}{|E|} \cdot \sum_{l \in E} L_l(\underline{f}) \qquad (N\text{-}1)$$

over $$\underline{f} \geq 0 \qquad (N\text{-}2)$$

subject to a) $$\sum_{a \in A} \phi_{v,a} \cdot f_a^{\pi} = \eta_v^{\pi} \cdot q^{\pi} \qquad \forall \pi \in \Pi, \forall v \in V \qquad (N\text{-}3)$$

b) $$f_a = \sum_{\pi \in \Pi} f_a^{\pi} \leq c_a \qquad \forall a \in A \qquad (N\text{-}4)$$

where $$\phi_{v,a} = \begin{cases} 1 & \text{(if arc } a \text{ leaves node } v\text{)} \\ -1 & \text{(if arc } a \text{ enters node } v\text{)} \\ 0 & \text{(otherwise)} \end{cases}$$

$$\eta_v^{\pi} = \begin{cases} 1 & \text{(if } v \text{ is an originating node of commodity } \pi\text{)} \\ -1 & \text{(if } v \text{ is a destination node of commodity } \pi\text{)} \\ 0 & \text{(otherwise)} \end{cases}$$

Equations (N-3) and (N-4) embody the flow conservation law and the capacity constraints, respectively. The arc-node flow representation [9] is employed on the commodity flow. f_a^{π} takes the amount of flow for commodity π over arc a. $L(\underline{f})$ gives an expected lost flow, while $L_l(\underline{f})$ is a lost flow due to a failure of link l. Based on the above formulation, the problem becomes a non-linear, non-smooth multicommodity flow problem with linear constraints.

In the calculation of an expected lost flow, we assume that it is possible to find a subset of affected virtual paths which completely fill up the bandwidth of a discovered restoration route. Since each link (i.e. Gbps - Tbps capacity for optical fiber) is shared among a number of commodities and each commodity is supposed to involve a large number of virtual paths, a single link will be shared by a significant number of virtual paths. This means that the bandwidth of a virtual path will be considerably smaller than a link bandwidth and the spare bandwidth over a restoration route. Therefore, even though any subset of virtual paths may not actually fit into the bandwidth of a restoration route, such a gap should be small, and the formulation based on this assumption would give a good approximation of the problem.

Now, $L_l(\underline{\mathbf{f}})$ is computed as follows: Consider a restoration process upon a failure of link l. Suppose that link l consists of two directed arcs, a and b, and assume that the restoration for each arc is processed in parallel. Let $\zeta_\beta^{(l,k)}(\underline{\mathbf{f}})$ denote the residual capacity of arc $\beta \in A \backslash \{a, b\}$ after the link restoration using the first k restoration routes for arcs a and b. It is given by the following recursive formula:

$$\zeta_\beta^{(l,0)}(\underline{\mathbf{f}}) = c_\beta - f_\beta$$

$$\zeta_\beta^{(l,k)}(\underline{\mathbf{f}}) = \zeta_\beta^{(l,k-1)}(\underline{\mathbf{f}}) \qquad \text{if } \beta \notin RP^{(a,k)} \cup RP^{(b,k)}$$

$$= \zeta_\beta^{(l,k-1)}(\underline{\mathbf{f}}) - r^{\alpha}_{RP^{(\alpha,k)}}(\underline{\mathbf{f}}) \qquad \text{if } \beta \in RP^{(\alpha,k)} \text{ for some } \alpha \in \{a, b\}$$

where $RP^{(\alpha,k)}$ is a set of directed arcs which are contained in the k-th shortest restoration route of arc $\alpha \in \{a, b\}$. Note that since a link is assumed to be bidirectional, the k-th shortest restoration routes for arcs a and b pass through the same links but in opposite directions. Thus, $RP^{(a,k)} \cap RP^{(b,k)} = \varnothing$. $r^{\alpha}_{RP^{(\alpha,k)}}(\underline{\mathbf{f}})$ is a restorable amount of flow of arc $\alpha \in l$ over the k-th restoration route. It is given by

$$r^{\alpha}_{RP^{(\alpha,k)}}(\underline{\mathbf{f}}) = \min\left[f_a - \sum_{t=1}^{k-1} r^{\alpha}_{RP^{(\alpha,t)}}(\underline{\mathbf{f}}), \min\{\zeta_\beta^{(l \ni a,k-1)}(\underline{\mathbf{f}}) : \beta \in RP^{(\alpha,k)}\} \right] \qquad (N\text{-}5)$$

Namely, $r^{\alpha}_{RP^{(\alpha,k)}}(\underline{\mathbf{f}})$ is calculated as the minimum of (a), the amount of flow which cannot be recovered over the first $k-1$ restoration routes, and (b), the restorable amount of flow over the k-th restoration route. The latter, (b), is equal to the minimum residual arc capacity over the route. Then, the total amount of restorable flow of arc $\alpha \in l$, $RST_\alpha(\underline{\mathbf{f}})$, is obtained by

$$RST_{\alpha}(\underline{\mathbf{f}}) = \sum_{k \in RP^{\alpha}} r^{\alpha}_{RP^{(\alpha,k)}}(\underline{\mathbf{f}})$$

where RP^{α} is a set of restoration routes for arc α. Finally, $L_l(\underline{\mathbf{f}})$ is computed by

$$L_l(\underline{\mathbf{f}}) = (f_a - RST_a(\underline{\mathbf{f}})) + (f_b - RST_b(\underline{\mathbf{f}})) \qquad (N\text{-}6)$$

The second formulation (SVPR-KSP-LINE2) describes the restoration flow constraints with the aid of the restoration path variables and the lost flow variables. This leads to a similar formulation to the SVPR-MF-LINE problem, but additional constraints are required in order to express the relationship given by (N-5). We call such constraints the *KSP constraints*. Although this formulation can eliminate the non-smoothness from the objective function, the constraint set becomes non-convex.

Given $G = (V, A, \underline{\mathbf{c}})$ and Q, **(SVPR-KSP-LINE2)**

Minimize $L \cdot |E| = \sum_{a \in A} t_a$

over $\underline{\mathbf{x}} = (x^{\pi}_p) \geq \mathbf{0}$, $\underline{\mathbf{r}} = \left(r^{a}_{RP^{(a,k)}}\right) \geq \mathbf{0}$, and $\underline{\mathbf{t}} = (t_a) \geq \mathbf{0}$

subject to a) $\sum_{p \in P^{\pi}} x^{\pi}_p = q^{\pi}$ $\quad \forall \pi \in \Pi$ $\quad (N\text{-}7)$

b) $f_a + z^{l}_a \leq c_a$ $\quad \forall a \in A, \forall l \in E_a$ $\quad (N\text{-}8)$

c) $-f_a + \sum_{k \in [RP^a]} r^{a}_{RP^{(a,k)}} + t_a = 0$ $\quad \forall a \in A$ $\quad (N\text{-}9)$

d) $r^{a \in l}_{RP^{(a,k)}} = \min\left\{ f_a - \sum_{i=1}^{k-1} r^{a}_{RP^{(a,i)}},\ \{\zeta^{(l,k-1)}_{RP^{(a,k)}_h} : h \in [RP^{(a,k)}]\} \right\}$

$\forall a \in A, \forall k \in [RP^a]$ $\quad (N\text{-}10)$

where $f_a = \sum_{\pi \in \Pi} \sum_{p \in P^{\pi}} \theta_{a,p} \cdot x^{\pi}_p$

$z^{l}_a = \sum_{\beta \in l} \sum_{k \in [RP^{\beta}]} \theta_{a, RP^{(\beta,k)}} \cdot r^{\beta}_{RP^{(\beta,k)}}$, and

$$\zeta_{RP_h^{(a,k)}}^{(l=\{a,b\},k-1)} = c_{RP_h^{(a,k)}} - f_{RP_h^{(a,k)}} - \sum_{i=1}^{k-1}\left(\theta_{RP_h^{(a,k)},RP^{(a,i)}} \cdot r^{a}_{RP^{(a,i)}} + \theta_{RP_h^{(a,k)},RP^{(b,i)}} \cdot r^{b}_{RP^{(b,i)}}\right)$$

The first three constraints (flow conservation law (N-7), capacity constraints (N-8) and restoration flow constraints (N-9)) are almost the same as those for the SVPR-MF-LINE problem, except that the restoration paths are restricted due to a hop limit (see Section 2.1.2.)[15]. $RP_h^{(a,k)}$ denotes the h-th arc in the k-th restoration path for arc a, and $[X] \equiv \{1, \ldots, |X|\}$. As before, $\zeta_\beta^{(l,k)}$ gives the residual capacity of arc β after restoration using the first k restoration paths upon a failure of link l. The KSP constraints are given by Equation (N-10). Each KSP constraint can be replaced by the following two inequalities:

$$r^{a \in l}_{RP^{(a,k)}} \le \min\left\{f_a - \sum_{i=1}^{k-1} r^{a}_{RP^{(a,i)}}, \{\zeta^{(l,k-1)}_{RP_h^{(a,k)}} : h \in [RP^{(a,k)}]\}\right\} \qquad \textit{(N-11)}$$

$$r^{a \in l}_{RP^{(a,k)}} \ge \min\left\{f_a - \sum_{i=1}^{k-1} r^{a}_{RP^{(a,i)}}, \{\zeta^{(l,k-1)}_{RP_h^{(a,k)}} : h \in [RP^{(a,k)}]\}\right\} \qquad \textit{(N-12)}$$

Note that Equation (N-11) is satisfied by constraints (N-8) and (N-9), and thus it is unnecessary.

Appendix O. SVPR-KSP-LINE: Solution approach - Modified flow deviation (MFD) method

The SVPR-KSP-LINE problem based on the first formulation (SVPR-KSP-LINE1) is a non-linear (non-smooth) programming problem with linear constraints. Such a problem can be addressed by an iterative method where a new feasible flow is found at each iteration, and the value of the objective function monotonically decreases. In order to maintain the feasibility, not only must a flow be a legal multicommodity flow[16] [9], but also the capacity constraints (N-4) must be

15. Thus, the first three constraints are same as the constraints for the SVPR-MF-LINE problem with an RP option.

checked at each iteration. The method also needs to find an initial feasible flow, which may be obtained through a linear programming algorithm. Instead of solving the above problem directly, we relax the capacity constraints as in [28]. If the total flow of each link is allowed to exceed its capacity, the capacity constraints can be removed from the formulation, and the iterative procedure is greatly simplified. Moreover, an initial solution is now obtained through a shortest path algorithm which is less computationally intensive than a linear programming approach especially for a large network.

Let $\Omega_A = \{\underline{f} : \underline{f}$ is a multicommodity flow ($\underline{f}$ satisfies (N-2) and (N-3))$\}$ and $\Omega_B = \Omega_A \cap \{\underline{f} :$ $\underline{f}$ satisfies (N-4)$\}$. The removal of the capacity constraints means expanding a feasible region from Ω_B to Ω_A. Then, in order to avoid the convergence at a point in $\Omega_A \backslash \Omega_B$, it is necessary that all local minima reside in Ω_B. This is accomplished by introducing a penalty function and modifying the definition of L as follows:

$$L^*(\underline{f}) = \frac{1}{|E|} \cdot \sum_{l \in E} L_l^*(\underline{f}) \qquad (O\text{-}1)$$

where

$$L_l^*(\underline{f}) = L_l(\underline{f}) + \omega \cdot \sum_{a \in A} \max\{0, f_a - c_a\}$$

Thus, if there is an arc $a \in A$ where the capacity constraint is violated, the excess flow $f_a - c_a$ is treated as a big lost flow by adding $\omega \cdot (f_a - c_a)$ to $L_l(\underline{f})$ for all $l \in E$.

The following Lemma gives a sufficient condition on the multiplier, ω, in order to guarantee the non-existence of local minima outside Ω_B. This suggests that, if an iterative method can always find a feasible descent direction, it can eventually find a point in Ω_B with a lower value of L^*, even if it starts from a point in $\Omega_A \backslash \Omega_B$.

Lemma O-1

Suppose $\omega \geq |A|$ and $\Omega_B \neq \varnothing$. Then, $L^*(\underline{f})$ has no local minima in $\Omega_A \backslash \Omega_B$.

(Proof)

16. In the minimum-cost multicommodity flow problems, a multicommodity flow is defined to be a flow satisfying the flow conservation law (N-3) and the non-negativity constraints (N-2) [9].

Let $\underline{f} \in \Omega_A \backslash \Omega_B$ and let A' ($A' \subseteq A$) be a set of the arcs whose flow exceeds the capacity in $\underline{f}$. Then, for an arc $a \in A'$ there is an $\underline{f}^{\pi}$-incrementing path [16], P_a, from j_a to k_a for some commodity $\pi \in \Pi$, where $a = (j_a, k_a)$. Otherwise, the excess flow $e_a = f_a - c_a$ cannot be rerouted, contradicting the assumption $\Omega_B \neq \varnothing$. Let FWD_a be a set of the forwarding arcs along P_a and RVS_a be a set of the reverse arcs along P_a. Define $i(P_a) \equiv \min\{i_b \mid b \in P_a\}$ where $i_b = c_b - f_b$ if $b \in FWD_a$ and $i_b = f_b^{\pi}$ if $b \in RVS_b$. Let $\varepsilon = \min\{e_a, f_a^{\pi}, i(P_a)\} > 0$ and $\underline{f}^*$ be the new flow which is obtained from $\underline{f}$ by decreasing f_a^{π} and f_b^{π} ($\forall b \in RVS_a$) by ε and increasing f_b^{π} ($\forall b \in FWD_a$) by ε. Apparently, $\underline{f}^* \in \Omega_A$. Note that the restorable flow against a failure of link $l \in E$ may be reduced by the increase of a flow in the forwarding arcs, namely at most by $|FWD_a| \cdot \varepsilon \leq (|A| - 1) \cdot \varepsilon$.

Let $E_{FWD_a} \equiv \{l \in E \mid (a_1 \in FWD_a) \cup (a_2 \in FWD_a), l = \{a_1, a_2\}\}$. Then, for a link $l \notin E_{FWD_a}$, we have

$$L_l^*(\underline{f}^*) \leq L_l^*(\underline{f}) - \omega \cdot \varepsilon + (|A| - 1) \cdot \varepsilon \qquad (O\text{-}2)$$

For a link $l \in E_{FWD_a}$, the increased amount of the flow in the forwarding arc may not be restored either. Thus,

$$L_l^*(\underline{f}^*) \leq L_l^*(\underline{f}) - \omega \cdot \varepsilon + (|A| - 1) \cdot \varepsilon + \varepsilon \qquad (O\text{-}3)$$

Note that the link containing the arc a is not in E_{FWD_a} and thus,

$$|E_{FWD_a}| < |E| \qquad (O\text{-}4)$$

Using the Equations (O-1) (O-2) (O-3) (O-4) and the relation $\omega \geq |A|$, we can derive

$$L^*(\underline{f}^*) \leq L^*(\underline{f}) - \varepsilon \cdot (|E| - |E_{FWD_a}|)/|E| < L^*(\underline{f})$$

This implies that there always exists a feasible descent direction at $\underline{f}$ if $\underline{f} \in \Omega_A \backslash \Omega_B$. Therefore, $L^*(\underline{f})$ has no local minima outside Ω_B.

(Q.E.D.)

Note that although the above proof assumes $w_s = 1/|E|$, any arbitrary weight on the failure events will produce the same result.

In summary, the survivable virtual path routing (SVPR-KSP-LINE1) problem is reformulated as follows, which we call the relaxed SVPR-KSP-LINE (RSVPR-KSP-LINE) problem:

Minimize	$L^*(\underline{\mathbf{f}}) = \frac{1}{\lvert E\rvert} \cdot \sum_{l \in E} L_l^*(\underline{\mathbf{f}})$	**(RSVPR-KSP-LINE)**
over	$\underline{\mathbf{f}} \geq \mathbf{0}$	
subject to	flow conservation law (N-3).	

Flow deviation method [22] [26] is an efficient algorithm for nonlinear multicommodity flow problems with convex constraint sets [13]. In order to calculate a search direction of the next iteration, the partial derivatives of the objective function are evaluated at the convergence point of each iteration. In the RSVPR-KSP-LINE problem, however, its objective function, L^*, is not differentiable everywhere. Although a descent direction at $\underline{\mathbf{f}}$ could be obtained if the objective function is differentiable at $\underline{\mathbf{f}}$, it is plausible that L^* is not differentiable at the convergence point of each iteration. This is because L^* is a piece-wise linear function, and a non-smooth point (kink) of such a function gives the lowest value in the search direction. Thus, the flow deviation method cannot be applied to the RSVPR-KSP-LINE problem directly.

An alternative way to find a feasible descent direction is to minimize the directional derivative over all feasible directions. Since any feasible direction at $\underline{\mathbf{f}}$ can be expressed by $\underline{\mathbf{v}} - \underline{\mathbf{f}}$ for some $\underline{\mathbf{v}} \in \Omega_A \backslash \{\underline{\mathbf{f}}\}$, the problem is formulated as a minimization of $DD(\underline{\mathbf{v}}, \underline{\mathbf{f}})$ over $\underline{\mathbf{v}} \in \Omega_A \backslash \{\underline{\mathbf{f}}\}$, where $DD(\underline{\mathbf{v}}, \underline{\mathbf{f}})$ is the directional derivative of $L^*(\underline{\mathbf{f}})$ along $\underline{\mathbf{v}} - \underline{\mathbf{f}}$. Due to non-smoothness of L^*, however, the directional derivative cannot be calculated just through a gradient. In order to overcome the difficulty, the following approximation, $DD^*(\underline{\mathbf{v}}, \underline{\mathbf{f}})$, is employed instead of $DD(\underline{\mathbf{v}}, \underline{\mathbf{f}})$:

$$DD^*(\underline{\mathbf{v}}, \underline{\mathbf{f}}) = \underline{g}^*(\underline{\mathbf{f}}) \bullet \frac{(\underline{\mathbf{v}} - \underline{\mathbf{f}})}{\|\underline{\mathbf{v}} - \underline{\mathbf{f}}\|}$$

where $\underline{g}^*(\underline{\mathbf{f}}) = [g_j^*(\underline{\mathbf{f}})] \quad (j \in A)$

$$g_j^*(\underline{\mathbf{f}}) = \begin{cases} g_j^+(\underline{\mathbf{f}}) & (\text{if} v_j - f_j > 0) \\ g_j^-(\underline{\mathbf{f}}) & (\text{if} v_j - f_j < 0) \end{cases}$$

$$g_i^+(\underline{\mathbf{f}}) \equiv \lim_{h \to 0^+} \{L^*(\underline{\mathbf{f}} + h \cdot \mathbf{e}_i) - L^*(\underline{\mathbf{f}})\} / h \text{ and}$$

$$g_i^-(\underline{\mathbf{f}}) \equiv \lim_{h \to 0^-} \{L^*(\underline{\mathbf{f}} + h \cdot \mathbf{e}_i) - L^*(\underline{\mathbf{f}})\} / h$$

where $\mathbf{e}_i$ is a unit vector with the i-th component equal to one. Note that $DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}})$ is a directional derivative of a piece-wise linear approximation of L^* using one-sided partial derivatives, $g_i^+(\underline{\mathbf{f}})$ and $g_i^-(\underline{\mathbf{f}})$.

Now, the problem is to minimize $DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}})$ over $\underline{\mathbf{y}} \in \Omega_A \setminus \{\underline{\mathbf{f}}\}$. Consider applying the flow deviation method for this new minimization problem. Again, the objective function $DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}})$ is not differentiable everywhere. However, the only non-differentiable region is $\Lambda = \{\underline{\mathbf{y}} \mid v_i = f_i \ (\exists i \in A)\}$. The partial derivative of $DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}})$ (except at the points in Λ) can be obtained as

$$\frac{\partial}{\partial v_i} DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}}) = \frac{g_i^*(\underline{\mathbf{f}})}{\sqrt{\sum_j (v_j - f_j)^2}} - \frac{\left[\sum_j g_j^*(\underline{\mathbf{f}}) \cdot (v_j - f_j)\right] \cdot (v_i - f_i)}{\left(\sum_j (v_j - f_j)^2\right)^{3/2}} \qquad (O\text{-}5)$$

When $v_i - f_i = 0$, we heuristically define $g_i^* = (g_i^+ + g_i^-)/2$ in Equation (O-5) in order to compute the partial derivative. According to our numerical experimentations, $v_i - f_i = 0$ seldom happens during the iterations.

$DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}})$ is not convex nor concave, so the flow deviation method gives only a stationary point of $DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}})$. With a good initial point $\underline{\mathbf{y}}^0$, however, it has a good chance of converging to a near-optimal point. If L^* is differentiable at $\underline{\mathbf{f}}$, it is known that $\underline{\mathbf{y}} - \underline{\mathbf{f}}$ gives a feasible descent direction [22] where $\underline{\mathbf{y}}$ is a shortest route flow obtained under the metric $\{\partial L^* / \partial f_i\}$, which is equal to $g_i^+ = g_i^- = (g_i^+ + g_i^-)/2$. Although L^* is not usually differentiable at the convergence point of each iteration, $\underline{\mathbf{y}}^0 - \underline{\mathbf{f}}$ would give a good approximation to a descent direction if $\underline{\mathbf{y}}^0$ is a shortest route flow under the metric $\{(g_i^+ + g_i^-)/2\}$. Therefore, using the $\underline{\mathbf{y}}^0$ as a starting point, the method could at least reach a local minimum with a negative directional derivative and possibly reach the minimum of $DD^*(\underline{\mathbf{y}}, \underline{\mathbf{f}})$.

The proposed algorithm is summarized as follows, which we call the *modified flow deviation* (*MFD*) method:

Procedure SVPR-KSP-LINE-MFD()

0. Find a feasible starting point $\underset{\sim}{f}^0 \in \Omega_A$. $\underset{\sim}{f}^0$ can be a shortest route flow if no apparent feasible solution is available. Let $n = 0$.
1. Through the following procedures, obtain $\underset{\sim}{y}^n$ such that $\underset{\sim}{y}^n - \underset{\sim}{f}^n$ would be the steepest descent direction.
   a) Find a feasible starting flow $\underset{\sim}{y}^0 \neq \underset{\sim}{f}^n$ in Ω_A. $\underset{\sim}{y}^0$ is a shortest route flow under the metric $\{(g_i^+ + g_i^-)/2\}$. Let $m = 0$.
   b) Obtain $\underset{\sim}{w}^m$ as a shortest route flow taking $\{\partial DD^*(\underset{\sim}{y}^m, \underset{\sim}{f}^n)/\partial v_i\}$ as an arc length.
   c) $\underset{\sim}{y}^{m+1} = (1-\mu^m)\underset{\sim}{y}^m + \mu^m \underset{\sim}{w}^m$
   where μ^m is the minimizer of $DD^*((1-\mu^m)\underset{\sim}{y}^m + \mu^m \underset{\sim}{w}^m), \underset{\sim}{f}^n)$ $(0 \le \mu^m \le 1)$.
   d) If $DD^*(\underset{\sim}{y}^m, \underset{\sim}{f}^n) - DD^*(\underset{\sim}{y}^{m+1}, \underset{\sim}{f}^n) < \delta$, then $\underset{\sim}{y}^n = \underset{\sim}{y}^{m+1}$ and go to Step 2. Otherwise, let m=m+1 and go back to Step 1-b).
2. $\underset{\sim}{f}^{n+1} = (1-\lambda^n)\underset{\sim}{f}^n + \lambda^n \underset{\sim}{y}^n$,
   where λ^n is the minimizer of $L^*((1-\lambda^n)\underset{\sim}{f}^n + \lambda^n \underset{\sim}{y}^n)$ $(0 \le \lambda^n \le 1)$.
3. If $L^*(\underset{\sim}{f}^{n+1}) = 0$ or $(L^*(\underset{\sim}{f}^n) - L^*(\underset{\sim}{f}^{n+1}))/L^*(\underset{\sim}{f}^n) < \varepsilon$, then stop. Otherwise, let n=n+1 and go back to Step 1.

The MFD method involves two iterative procedures, outer iteration (Step 1~3) and inner iteration (Step 1-b~d). The inner iteration seeks the steepest descent direction at each loop of the outer iteration. Using the result of the inner iteration, the outer iteration proceeds towards a minimum point. A golden section search method [54] is used to implement the line search in Step 1-c) and Step 2).

Implementation Issues

Since $L^*(\underset{\sim}{f})$ and $g^*(\underset{\sim}{f})$ have no closed form solutions, they can be calculated only numerically. The calculation of $L_l^*(\underset{\sim}{f})$ involves repeated application of the shortest path algorithm to find the successively shortest restoration routes over the residual network. Assuming the Dijkstra's algorithm with binary heap [14], it requires $O(A^2 \log V)$ time in the worst case since at least one arc is removed from the residual network at each application of the shortest path algorithm. Then, the total running time for $L^*(\underset{\sim}{f})$ and $g^*(\underset{\sim}{f})$ goes up to $O(A^3 \log V)$ and $O(A^4 \log V)$, respectively. Although their complexities are polynomial, they slow down the algorithm considerably since these operations are invoked at every outer iteration. In order to reduce the computational time, a precalculated table of the restoration routes is used in our implementation. Although the size of such a table grows exponentially as the size of the network increases, our preliminary study

shows that it is enough to consider the first 30 shortest restoration routes to obtain a very close approximation of $L^*(\underline{f})$. This strategy reduces the computational complexities of $L^*(\underline{f})$ and $\underline{g}^*(\underline{f})$ down to $O(A)$ and $O(A^2)$, respectively. Table 0-1 shows required CPU time for the MFD method. The same test conditions as those used in the SVPR-MF problems are employed in this experiment. The result indicates that the MFD method takes very short time and is applicable to a network with rapidly changing traffic demand patterns.

The MFD method employs several optimization parameters. Using two sample networks, we have examined the relationship of the optimization parameter values to the performance of the algorithm. Tens of demand patterns have been examined for each model to see the behavior of the algorithm in a lightly loaded situation as well as a heavily loaded case. Some are uniformly distributed and some are randomly distributed.

There are three parameters which might affect the performance of the algorithm: the stopping conditions, ε and δ, and the step size, h. The stopping conditions are employed in the termination tests in the outer and inner iterations, while the step size is used to numerically calculate the partial derivatives, $g_i^*(\underline{f})$. It has been found that the stopping condition ε does not have much to do with the performance of the algorithm as long as it is sufficiently small. Furthermore, when the load is light, the selection of the parameters does not make significant difference in the convergence speed as long as they are reasonably small. Since the optimal solution set is not a singleton in this case, the procedure can find some point in the set rather easily regardless of the parameter values. In general, the convergence point depends on the choice of the parameters, but this is not a problem from the survivability point of view, since a minimum expected lost flow is attained.

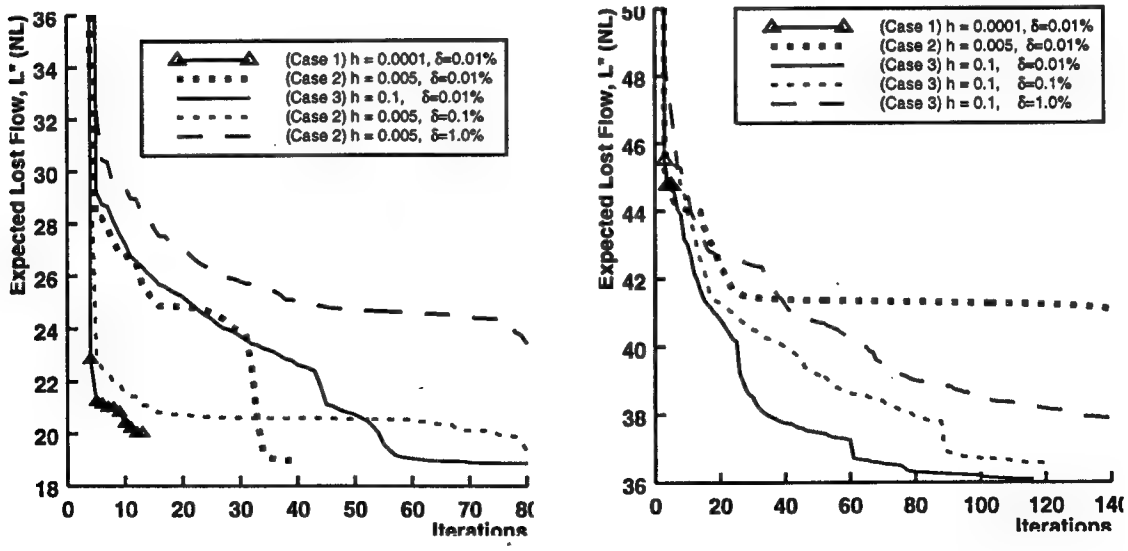
In a heavily loaded situation, however, the convergence speed greatly depends on the selection of the two parameters, h and δ. Figure 0-1 illustrates typical transitions of L^* over iterations for various values of h and δ. The results on the NJ-LATA and US-WAN sample networks are

Table 0-1. Required CPU time for the MFD method

Network Model	NJ-LATA	(12,50)	(24,60)	US-WAN
100% RNL with 30 random variations	1.96 s	5.82 s	2.53 s	17.6 s
102.5% RNL with no variation	2.05 s	9.53 s	11.4 s	46.2 s

reported. The latter case has a heavier offered load than the former. Three curves are plotted for each case representing small (case 1), medium (case 2) and relatively large (case 3) values of h. Two additional curves are depicted for different values of δ. Note that the following discussion on the convergence is a generalization of the results of many experiments with different traffic patterns. There is no definite rule, and the convergence pattern is different in each case.

Generally speaking, a smaller step size, h, is thought to be desirable since it gives a more precise approximation of the partial derivatives, and this is the case when the load is not so heavy. In a heavily loaded situation, however, the procedure often stops prematurely after a sharp decrease in the first few iterations (Figure 0-1-a: case 1). This phenomenon becomes more prominent when the load is too heavy (Figure 0-1-b). The premature convergence is due to a lack of global information on the change of L^*. A search direction obtained through a smaller step size might be descent only in the neighborhood around $\underline{f}$. This area is usually small for heavily loaded networks since a small flow change at one link can readily affect the restorability of some other links. Since the minimum point along the search direction often falls near $\underline{f}$, enough progress cannot be made at each iteration. As a result, a decrease of L^* is too small to pass the termination test, and premature termination follows.



(a) NJ-LATA Sample Network (b) US-WAN Sample Network

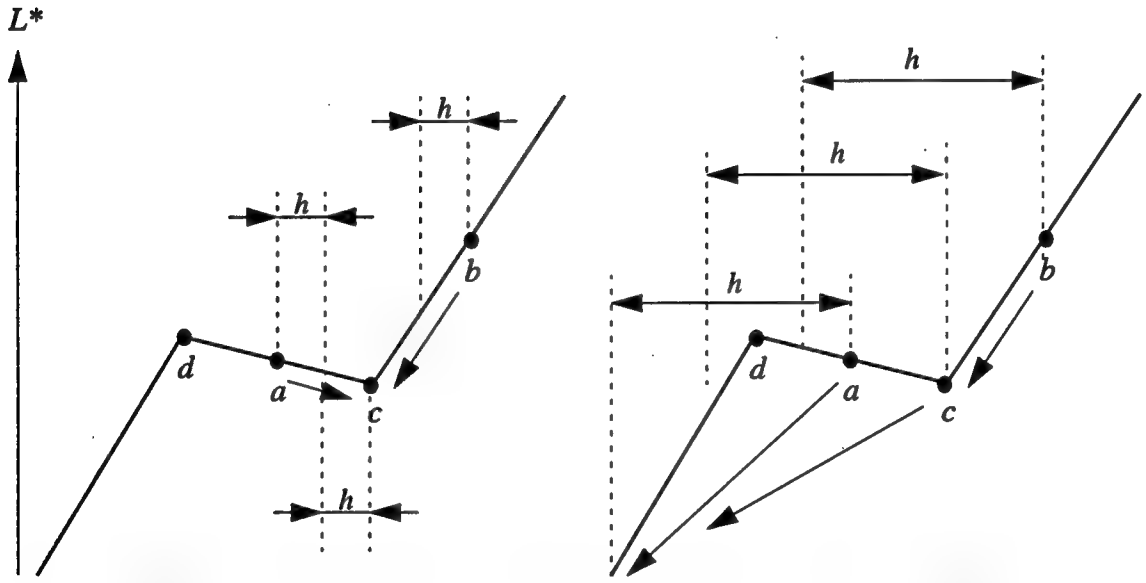
Figure 0-1. Typical transition of L* over iterations

Furthermore, the accuracy of the linear approximation of a nonsmooth function might degrade in a heavy load, since lots of turning points (kinks) are expected to exist in a small region. It is well-known that the steepest descent method could fail around a kink if it is applied to a non-smooth optimization [49]. This is because the linear approximation based on a gradient may not be able to take into account a large change in the objective function value at a non-smooth point, even if the point is located in the neighborhood (see Figure 0-2). Premature convergence with a smaller h is also explained by this inaccuracy of the linear approximation.

A larger step size is generally considered undesirable since it may mislead to a false search direction due to a possible error in obtaining derivatives. In fact, the procedure converges very slowly as shown in Figure 0-1. However, as the load grows, only a relatively large h can approach an optimal point (Figure 0-1-b). Since the next direction is obtained by seeking a broader range around $\underline{\mathbf{f}}$, the minimum point along the search direction tends to reside away from $\underline{\mathbf{f}}$. Namely, the procedure can find a longer descent slope, although it might not be the steepest descent direction at $\underline{\mathbf{f}}$. Consequently, it usually produces sufficient decrease to prevent premature convergence even in a heavily loaded network, although the attained improvement rate might be low. In other words, the linear approximation of L^* with a large step size is not precise locally but gives a global view. This approximation works well in heavily loaded situations since the effect of dense non-smooth points is smoothed out over a relatively large neighborhood. Premature convergence at a kink can be avoided as shown in the example of Figure 0-2-b. In addition, a search direction obtained through a larger step size might find a direction with a very long downhill, and this results in an occasional large reduction of L^*.

In summary, the best choice of step size greatly depends on the traffic load. A small h gives the best result in a lightly loaded case, while with a heavy load it shows a quick decrease but results in a premature convergence. Thus, a larger step size is required as the load grows, but it should not be unnecessarily large in order to avoid longer iterations (see Figure 0-1-a). Since load information is not usually obtainable before starting the procedure, it is very hard to choose a single step size to give the best result. One possible guideline is to change the value over the iterations. The procedure starts with a smaller step size to achieve fast convergence for a lightly loaded case or to induce a quick decrease of L^*. Then, it gradually increases the step size to obtain a better solution for a heavily loaded case. Based on the experiments, this strategy generally works well, although in some cases more iterations are required.

The stopping condition, δ, influences not only the required number of inner iterations but also that of outer iterations. Obviously, a smaller number of inner iterations are necessary for a larger δ. On the other hand, a smaller δ requires fewer outer iterations since a steeper downhill could be found at each iteration. Considering the fact that the outer iteration is more computationally expensive than the inner iteration, it is better to use a smaller value for δ. Furthermore, a large δ may cause premature convergence since the minimization process of *DD** can terminate before finding a descending direction. The value of δ, however, should not be unnecessarily small. The objective function of the inner iteration is just an approximation. Fine tuning on such a function might not always lead to a better solution unless the approximation is very precise. In fact, a smaller δ occasionally fails to find a better direction. In the example shown in Figure 0-1-a, the curve with δ=0.01 shows slower improvement than that with δ=0.1 at the early stage of the iterations. According to our experiments, δ less than 0.1% usually works well.



(a) MFD with a small step size

(b) MFD with a large step size

Figure 0-2. Avoidance of premature convergence around a kink

With a small *h*, the MFD procedure may stall at a kink *c* if started from point *a* or *b* (Figure (a)). This is because a linear approximation by a small *h* around point *c* cannot take into account a large change of the objective function value at *d*. On the other hand, the procedure can get out of the kink with a large *h*, by one step from *a*, or by two steps from *b* (Figure b).

Appendix P. SVPR-KSP-LINE: Solution approach - Lagrangian relaxation method

As will be discussed in Section 4.4.4., the MFD method can find a near-optimal point when the network load is not heavy. Considering the speed of the solution procedure, the proposed algorithm is very practical. However, the optimality degrades as the load grows further. Premature convergence around a kink causes this problem, and a larger step size could avoid the problem as discussed above. Since more non-smooth points are introduced at a higher load, however, premature convergence becomes inevitable even with a large h, causing the degradation in optimality.

The second formulation (SVPR-KSP-LINE2) can eliminate the non-smoothness from the objective function. Thus, it is expected to reach a better solution at a higher load. However, the KSP constraints (N-10) are too complicated to be tackled directly. In order to overcome the difficulties, we apply the Lagrangian relaxation method. By dualizing complicated constraints, a Lagrangian relaxation method could simplify the solution procedure (see Appendix S for details). The technique would also provide a way to find a tighter lower bound which is useful in evaluating the proposed heuristics. A major disadvantage of this approach is that it would require considerably more computation time due to the increased number of constraints.

The Lagrangian relaxation technique seeks a better solution by iteratively solving a Lagrangian subproblem and a Lagrangian dual problem. The problem structure heavily depends on the choice of a constraint set to be relaxed. Out of several possible options, we relax only the KSP constraints. There are several advantages in this choice. First of all, this has led to the best solution in terms of optimality based on our preliminary experiments. Secondly, a primally feasible solution can be readily found since the commodity flow acquired in the Lagrangian subproblem is also primally feasible[17]. Finally, a similar solution procedure to the SVPR-MF-LINE problem can be adopted by relaxing the KSP constraints (Equation (N-12)).

For a given multiplier $\tilde{\rho} = (\rho_{a,k}) \geq \mathbf{0}$, define a Lagrangian function $\Im_{\rho}(\underline{x}, \underline{r}, \underline{t})$ as:

17. A Lagrangian relaxation method requires a mechanism to find a primally feasible solution since it generally produces only a dually feasible solution. If a feasible commodity flow is at hand, then a feasible restoration flow and a feasible lost flow can be readily computed through Equations (N-9) and (N-10).

$$\Im_{\rho}(\underline{x}, \underline{r}, \underline{t}) \equiv \sum_{a \in A} t_a + \sum_{a \in A} \sum_{k=1}^{|RP^a|} \rho_{a,k} \cdot \left(\min \left\{ f_a - \sum_{i=1}^{k-1} r^a_{RP^{(a,i)}}, \{ \zeta^{(l \ni a, k-1)}_{RP^{(a,k)}_h} : h \in [RP^{(a,k)}] \} \right\} - r^a_{RP^{(a,k)}} \right)$$

$\rho_{a,k}$ is a Lagrangian multiplier corresponding to the constraint $KSP^{(a,k)}$ where $KSP^{(a,k)}$ is the KSP constraint for arc a and the k-th restoration path of the arc. Now, a Lagrangian subproblem can be defined as follows:

Minimize $\Im_{\rho}(\underline{x}, \underline{r}, \underline{t})$ **(SVPR-KSP-LINE-LAG)**

over $\underline{x}, \underline{r}, \underline{t} \geq 0$

subject to

a) $\sum_{p \in P^{\pi}} x^{\pi}_p = q^{\pi}$ $\forall \pi \in \Pi$ *(P-1)*

b) $f_a + z^l_a \leq c_a$ $\forall a \in A, \forall l \in E_a$ *(P-2)*

c) $-f_a + \sum_{k \in [RP^a]} r^a_{RP^{(a,k)}} + t_a = 0$ $\forall a \in A$ *(P-3)*

Let $\Im(\tilde{\rho})$ be the solution to the above minimization problem. Namely,

$\Im(\tilde{\rho}) = \min\{\Im_{\rho}(\underline{x}, \underline{r}, \underline{t}) :$ (P-1), (P-2) and (P-3)$\}$

Now the Lagrangian dual problem is stated as the following optimization problem:

Maximize $\Im(\tilde{\rho})$ **(SVPR-KSP-LINE-LAG-DUAL)**

over $\tilde{\rho}$

subject to $\tilde{\rho} \geq 0$

By weak duality [4], $\min\{L \cdot |E|\} \geq \max\{\Im(\tilde{\rho}) : \tilde{\rho} \geq 0\}$. The subgradient method described in Appendix S.2 is employed to update the multipliers which could give a tighter lower bound.

Now it can be easily shown that the multiplier is monotonically non-decreasing. Thus, $\tilde{\rho}$ should be initialized to $\mathbf{0}$ so that a multiplier could reach any value in its domain. In brief, the proposed solution procedure takes the following steps:

procedure SVPR-KSP-LINE-LAG() (brief description)

0. Set $\tilde{\rho} = \mathbf{0}$.
1. Solve the Lagrangian subproblem (SVPR-KSP-LINE-LAG).
2. Based on the feasible commodity flow assignment $\underset{\sim}{x}$ obtained in Step 1, calculate a feasible restoration flow $\underset{\sim}{r}$ and a lost flow $\underset{\sim}{t}$ through Equations (N-9) and (N-10).
3. Perform a termination test.
4. Solve the Lagrangian dual problem (SVPR-KSP-LINE-LAG-DUAL) by the subgradient method. Update the Lagrangian multiplier $\tilde{\rho}$. Go to Step 1.

The above procedure can be interpreted as follows: The solution of the Lagrangian subproblem at the first iteration is same as the optimum of the SVPR-MF-LINE problem with an RP option since $\tilde{\rho} = \mathbf{0}$. Due to the difference of the restoration protocols, the optimum of the SVPR-KSP-LINE problem would be somewhat different from this initial solution. Thus, starting from the solution to the MF-LINE-based network, the algorithm attempts to adjust a commodity flow to accommodate this difference. From the viewpoint of the problem formulation, this difference comes from the KSP constraints. Now, if a constraint $KSP^{(a,k)}$ is violated, the subgradient method increases the value of $\rho_{a,k}$ in proportion to the degree of the violation. In other words, a penalty is imposed on violated KSP constraints in the Lagrangian subproblem and is applied in the next iteration. With the aid of this penalty (Lagrangian multiplier), the algorithm updates a flow towards the optimum for the KSP-based network. The penalty is refined over iterations, and the procedure *SVPR-KSP-LINE-LAG* would eventually find a near-optimal solution to the SVPR-KSP-LINE problem.

Now consider the Lagrangian subproblem (SVPR-KSP-LINE-LAG). First of all, define $\Im_{\rho}(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t}, \underset{\sim}{y})$ as follows:

$$\Im_\rho(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t}, \underset{\sim}{y}) \equiv \sum_{a \in A} t_a + \sum_{a \in A} \sum_{k=1}^{|RP^a|} \rho_{a,k} \cdot \left(\left(y^0_{a,k} \cdot \left(f_a - \sum_{i=1}^{k-1} r^a_{RP^{(a,i)}} \right) + \sum_{h=1}^{|RP^{(a,k)}|} y^h_{a,k} \cdot \zeta^{(l \ni a, k-1)}_{RP^{(a,k)}_h} \right) - r^a_{RP^{(a,k)}} \right)$$

where $y^h_{a,k} = 0$ or $1 \quad \left(a \in A, k \in |RP^a|, h \in \{0, \ldots, |RP^{(a,k)}|\} \right)$ and $\sum_{h=0}^{|RP^{(a,k)}|} y^h_{a,k} = 1$

Then, $\Im_\rho(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t}, \underset{\sim}{y})$ can be transformed into the following:

$$\begin{aligned}
\Im_\rho(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t}, \underset{\sim}{y}) = & \sum_{a \in A} \left(1 + \sum_{k=1}^{|RP^a|} \rho_{a,k} \cdot y^0_{a,k} - \sum_{\beta \in A} \sum_{k=1}^{|RP^\beta|} \sum_{h=1}^{|RP^{(\beta,k)}|} \theta^{=}_{a, RP^{(\beta,k)}_h} \cdot \rho_{\beta,k} \cdot y^h_{\beta,k} \right) \cdot f_a \\
& + \sum_{a \in A} \sum_{k \in [RP^a]} \left[-1 - \rho_{a,k} - \sum_{i=k+1}^{|RP^a|} \rho_{a,i} \cdot y^0_{a,i} \right. \\
& \quad - \left(\sum_{i=k+1}^{|RP^a|} \sum_{h=1}^{|RP^{(a,i)}|} \theta_{RP^{(a,i)}_h, RP^{(a,k)}} \cdot \rho_{a,i} \cdot y^h_{a,i} \right. \qquad (l = \{a, b\}) \\
& \quad \left. \left. + \sum_{i=k+1}^{|RP^b|} \sum_{h=1}^{|RP^{(b,i)}|} \theta_{RP^{(b,i)}_h, RP^{(a,k)}} \cdot \rho_{b,i} \cdot y^h_{b,i} \right) \right] \cdot r^a_{RP^{(a,k)}} \\
& + \sum_{a \in A} \sum_{k=1}^{|RP^a|} \sum_{h=1}^{|RP^{(a,k)}|} \rho_{a,k} \cdot y^h_{a,k} \cdot c_{RP^{(a,k)}_h}
\end{aligned}$$

where $\theta^{=}_{a,b}$ takes one if $a = b$ and zero otherwise. Define

$$F_a \equiv 1 + \sum_{k=1}^{|RP^a|} \rho_{a,k} \cdot y^0_{a,k} - \sum_{\beta \in A} \sum_{k=1}^{|RP^\beta|} \sum_{h=1}^{|RP^{(\beta,k)}|} \theta^{=}_{a, RP^{(\beta,k)}_h} \cdot \rho_{\beta,k} \cdot y^h_{\beta,k}$$

$$\begin{aligned}
R^a_k \equiv & -1 - \rho_{a,k} - \sum_{i=k+1}^{|RP^a|} \rho_{a,i} \cdot y^0_{a,i} \qquad (l = \{a, b\}) \\
& - \left(\sum_{i=k+1}^{|RP^a|} \sum_{h=1}^{|RP^{(a,i)}|} \theta_{RP^{(a,i)}_h, RP^{(a,k)}} \cdot \rho_{a,i} \cdot y^h_{a,i} + \sum_{i=k+1}^{|RP^b|} \sum_{h=1}^{|RP^{(b,i)}|} \theta_{RP^{(b,i)}_h, RP^{(a,k)}} \cdot \rho_{b,i} \cdot y^h_{b,i} \right)
\end{aligned}$$

$$C \equiv \sum_{a \in A} \sum_{k=1}^{|RP^a||RP^{(a,k)}|} \sum_{h=1} \rho_{a,k} \cdot y_{a,k}^{h} \cdot c_{RP_h^{(a,k)}}$$

Then,

$$\Im_\rho(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t}, \underset{\sim}{y}) = \sum_{a \in A} F_a \cdot f_a + \sum_{a \in A} \sum_{k \in [RP^a]} R_k^a \cdot r_{RP^{(a,k)}}^a + C$$

$$= \sum_{a \in A} F_a \cdot t_a + \sum_{a \in A} \sum_{k \in [RP^a]} \tilde{R}_k^a \cdot r_{RP^{(a,k)}}^a + C$$

where $\tilde{R}_k^a = F_a + R_k^a$ Now, let $\Im(\tilde{\rho}, \underset{\sim}{y})$ denote the minimum of $\Im_\rho(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t}, \underset{\sim}{y})$ for given $\underset{\sim}{y}$ and $\tilde{\rho}$. Then, $\Im(\tilde{\rho}, \underset{\sim}{y})$ can be obtained by solving the following optimization problem (SVPR-KSP-LINE-LAG2):

Minimize $\sum_{a \in A} F_a \cdot t_a + \sum_{a \in A} \sum_{k \in [RP^a]} \tilde{R}_k^a \cdot r_{RP^{(a,k)}}^a + C$ **(SVPR-KSP-LINE-LAG2)**

over $\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t} \geq 0$

subject to a) $\sum_{p \in P^\pi} x_p^\pi = q^\pi$ $\forall \pi \in \Pi$

b) $f_a + z_a^l \leq c_a$ $\forall a \in A, \forall l \in E_a$

c) $-f_a + \sum_{k \in [RP^a]} r_{RP^{(a,k)}}^a + t_a = 0$ $\forall a \in A$

Maximize $\sum_{\pi \in \Pi} q^\pi \cdot \sigma^\pi + \sum_{a \in A} c_a \cdot \left(\sum_{l \in E_a} \mu_a^l \right)$ **(SVPR-KSP-LINE-LAG2-DUAL)**

over σ^π, w_a unrestricted, $\mu_a^l \leq 0$

subject to i) $\sum_{a \in p} \left(w_a - \sum_{l \in E_a} \mu_a^l \right) \geq \sigma^\pi$ $\forall p \in P^\pi, \forall \pi \in \Pi$

ii) $\sum_{b \in RP^{(a,k)}} \mu_b^{l \ni a} + w_a \leq \tilde{R}_k^a$ $\forall k \in [RP^a], \forall a \in A$

iii) $w_a \leq F_a$ $\quad\quad\quad\quad \forall a \in A$

The (SVPR-KSP-LINE-LAG2-DUAL) shows the dual formulation. Due to the similarity of their problem structures, the above problem can be approached in a similar fashion to the SVPR-MF-LINE problem with an RP option. The pricing-out operation and dual feasibility test must be slightly modified according to the changes in the dual constraints.

Let $KSP^{h}_{a,k}$ denote the h-th term[18] in the argument of the min function in the constraint $KSP^{(a,k)}$ $\left(h \in \{0, 1, \ldots, |RP^{(a,k)}|\}\right)$ (refer to Equation (N-10)). Now, suppose that the following inequality holds:

$$KSP^{h^*}_{a,k} \leq KSP^{\forall h \neq h^*}_{a,k}$$

Let the indicator variables $y^{h}_{a,k}$ take the following values on such an occasion:

$$y^{h = h^*}_{a,k} = 1 \text{ and } y^{h \neq h^*}_{a,k} = 0$$

Then, apparently $\Im_{\rho}(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t}, \underset{\sim}{y}) = \Im_{\rho}(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t})$. Therefore, the following minimization problem solves the Lagrangian subproblem:

Minimize	$\Im(\tilde{\rho}, \underset{\sim}{y})$	**(SVPR-KSP-LINE-LAG3)**
over	$\underset{\sim}{y}$	
subject to	$y^{h}_{a,k} = 1$ if h satisfies $KSP^{h}_{a,k} \leq KSP^{\forall i \neq h}_{a,k}$	
	$y^{h}_{a,k} = 0$ otherwise.	

A local minimum for the above minimization problem can be found by the procedure, *SVPR-KSP-LINE-LAG-local.*

18. It is assumed to be counted from zero.

procedure SVPR-KSP-LINE-LAG-local($\underset{\sim}{y}$)

1. For a given $\underset{\sim}{y}$, obtain $\Im(\tilde{\rho}, \underset{\sim}{y})$ by solving the problem (SVPR-KSP-LINE-LAG2).
2. For all KSP constraints $KSP^{(a,k)}$ with $\rho_{a,k} > 0$, check if $KSP^{h}_{a,k} \leq KSP^{\forall i \neq h}_{a,k}$ with $y^{h}_{a,k} = 1$. If true, go to Step 4.
3. Update $\underset{\sim}{y}$ so that the above condition is met. Go to Step 1.
4. If there is $i \neq h$ with $KSP^{h}_{a,k} = KSP^{i}_{a,k}$ and $y^{h}_{a,k} = 1$ for some constraint $KSP^{(a,k)}$ with $\rho_{a,k} > 0$, then check if $y^{i}_{a,k} = 1$ won't reduce the objective function value. If the value decreases, update $\underset{\sim}{y}$ and go to Step 2.
5. Return with the minimum value and the corresponding value of $\underset{\sim}{y}$ and $\underset{\sim}{x}$.

Note that at each iteration, the objective function value always improves since $\rho_{a,k} > 0$ and $KSP^{\exists i \neq h}_{a,k} < KSP^{h}_{a,k}$. Therefore, the algorithm terminates with a finite number of iterations since the combination of $\underset{\sim}{y}$ is finite.

The minimum calculated in the above manner is a local minimum and is not guaranteed to be a global one[19]. Therefore, the obtained solution is not assured to be a lower bound of the optimum of the SVPR-KSP-LINE problem. Although a global optimum may not be acquired in general, a random search technique of the global optimization can be applied to find an estimate of a lower bound [77]. For a given multiplier, the technique randomly selects multiple starting points $\underset{\sim}{y}$ and obtains multiple local minima. By increasing the number of sample points, the lowest value among the local minima would converge to a global minimum in a probabilistic sense [77]. The procedure *SVPR-KSP-LINE-LAG-global* embodies this random search technique:

procedure SVPR-KSP-LINE-LAG-global(bestLBD,bestUBD)

The arguments bestLBD and bestUBD contain the best lower and upper bounds ever found.

0. Set $j = 1$ and $LBD = \infty$.
1. Randomly generate $\underset{\sim}{y}$.
2. Call *SVPR-KSP-LINE-LAG-local*($\underset{\sim}{y}$). Obtain a local minimum, $Lmin_j$, a corresponding commodity flow assignment $\underset{\sim}{x}$, and an updated $\underset{\sim}{y}$. If $LBD > Lmin_j$, then set $LBD = Lmin_j$.
3. Obtain an expected lost flow, UBD_j, based on $\underset{\sim}{x}$ via Equation (N-1). If $bestUBD > UBD_j$, then set $bestUBD = UBD_j$.

19. Since the Lagrangian subproblem (SVPR-KSP-LINE-LAG) is a concave minimization problem with a convex constraint set, it can yield many local minima at its extreme points.

4. Perform termination test.
   a) If $j < Min\_Global\_Iter$, then $j \leftarrow j + 1$ and go to Step 1.
   b) If $LBD < bestLBD$, go to Step 5.
   c) If $j \geq Max\_Global\_Iter$, then go to Step 5.
   d) Otherwise, $j \leftarrow j + 1$ and go to Step 1.
5. Return with *bestUBD*, *LBD*, and $\underset{\sim}{y}$ which gives *LBD*.

In the above procedure, at least *Min_Global_Iter* random samples are generated, and the procedure stops with at most *Max_Global_Iter* samples. The global optimization procedure *SVPR-KSP-LINE-LAG-global* involves extensive computation if a good estimate of the lower bound is necessary. The values of *Min_Global_Iter* and *Max_Global_Iter* must be sufficiently large[20] for this purpose.

Nonetheless, the approach based on the SVPR-KSP-LINE2 formulation is very useful in obtaining a good approximation of an optimal solution. If the objective is to find a primal solution, not a lower bound, then a good estimate of a globally optimal solution is not necessary in the Lagrangian subproblem. Therefore, *Min_Global_Iter* and *Max_Global_Iter* need not have large values. In our experiments, they are set to 20 and 30, respectively. Furthermore, the global optimization procedure is not even necessary at all iterations. Instead, it would suffice to call the process only occasionally, say only when a local minimum becomes too close to or beyond the best upper bound found so far[21]. As the procedure converges, it is expected that a commodity flow assignment tends to be near-optimal. This procedure typically produces a better solution than the MFD method since it is immune to the issue arising from the kinks.

---

20. In order to reduce the sample points, a procedure proposed by Golden et.al. could be applied [32] [33]. It is based on a statistical extreme-value theory: Assuming randomness of the samples and a Weibull distribution for their limiting distribution, it calculates a point estimate and a confidence interval for a global optimum. The above assumption is empirically justified by applying it to a large-scale travelling salesman problem.

21. Namely, a global optimization procedure is not invoked until a Lagrangian multiplier approaches a near-optimal point in the Lagrangian dual problem. If a local minimum becomes too close to or beyond an upper bound estimate, then either a Lagrangian multiplier is near-optimal in the Lagrangian dual problem, or the multiplier deviates from the optimum due to a non-trivial gap between the local and global minima in the Lagrangian subproblem. In the former case, we can terminate the procedure. In the latter case, however, it is necessary to find a better local minimum to adjust the multiplier in the next iteration of the Lagrangian dual problem. In either case, a global optimization procedure should be invoked to check if adjustment of a multiplier is necessary or not. However, a thorough random search is not required on this occasion. If the gap is large, then a better local minimum can typically be found in a few random searches. Even if a global optimum may not be found, it is sufficient to perform a few random searches in order to adjust a multiplier. Thus, *Min_Global_Iter* and *Max_Global_Iter* need not have large values.

The proposed algorithm is summarized as follows, which we call the *LAG* method:

procedure SVPR-KSP-LINE-LAG()

0. Initialize $\underset{\sim}{y}$. Set $\tilde{\rho} = 0$, $UBD = \infty$, $LBD = 0$ and $j = 1$.
1. Call *SVPR-KSP-LINE-LAG-local*($\underset{\sim}{y}$). Obtain a local minimum, $Lmin_j$, a corresponding commodity flow assignment $\underset{\sim}{x}$, and an updated $\underset{\sim}{y}$.
2. Obtain an expected lost flow, UBD_j, based on $\underset{\sim}{x}$ via Equation (N-1).
   If $UBD > UBD_j$, then set $UBD = UBD_j$.
3. If $(UBD - Lmin_j) / (UBD) < \varepsilon_G$, then
   a) Call *SVPR-KSP-LINE-LAG-global(LBD,UBD)* and obtain an estimated lower bound $ELBD_j$ and the best upper bound $BUBD_j$ encountered during the procedure.
   b) If $UBD > BUBD_j$, then set $UBD = BUBD_j$.
   c) If $LBD < ELBD_j$, then set $LBD = ELBD_j$.
4. Stop if either of the following hold.
   a) $(UBD - LBD) / (UBD) < \varepsilon_B$
   b) The step length given by Equation (S-7) on page 168 is less than ε_S.
5. Solve the Lagrangian dual problem through the subgradient method. Calculate the subgradient and update the Lagrangian multiplier and step length.
6. $j \leftarrow j + 1$. Go to Step 1.

Appendix Q. Quadratic Shortest Path (QSP) Algorithm

The quadratic shortest path (QSP) algorithm solves the following problem:

> Find the shortest path from a source S to a destination D in the bidirectional network G where each path is subject to two types of arc cost: The *independent arc cost*, $\{c_a\}$, is imposed if a path goes through arc a, and the non-positive *mutual arc cost*, $\{m_{a,l}\}$, is added if a path contains both arc a and link l.

In the SCFA-MF-ETE problem, c_a and $m_{a,l}$ equal $d_a + w^{\pi, l \ni a}$ and $-\mu_a^l$, respectively, while in

the SVPR-MF-ETE problem, the two costs are equal to $w^{\pi, l \ni a} - \sum_{l \in E} \mu_a^l$ and μ_a^l, respectively. The proposed QSP algorithm works if the following three conditions are met, which can be proven in the cases of the SCFA-MF-ETE and SVPR-MF-ETE problems:

$c_a \geq 0$ for $\forall a \in A$,

$m_{a,l} \leq 0$ for $\forall a \in A$ and $\forall l \in E_a$,

$m_{a, l \ni a} = 0$, and

$c_a' = c_a + \sum_{l \in E_a} m_{a,l} \geq 0$ for $\forall a \in A$.

The algorithm stems from the idea of the branch and bound technique [81]. It searches all possible paths via a depth-first search. Along with the search, unnecessary paths are eliminated from consideration by using the best upper bound information obtained so far.

QSP algorithm

procedure main()

1) Find the shortest paths from all nodes to D, using $\{c_a'\}$ as an arc cost. Let $d(v)$ denote the minimum cost from a node v to D.
2) Let P be the shortest path from S to D found in the previous step. Set
$$UB = \sum_{a \in P} c_a + \sum_{a \in P} \sum_{l \in P} m_{a,l}.$$
3) Mark S as 'visited'. Mark all other nodes as 'unvisited'.
4) Call *visit* (S, $\varnothing$, 0)
5) The solution is given by P (the shortest path) and UB (its cost).

procedure visit (vertex, path, cost)

1) Mark *vertex* as 'visited'.
2) For all arc a adjacent to *vertex*,

2-1) If $end\_node(a)$ is marked as 'visited', continue to the next arc.

2-2) If $new\_cost + d(end\_node(a)) \geq UB$, continue to the next arc, where new_cost is given by
$$new\_cost = cost + c_a + \sum_{b \in path} m_{b, l \ni a} + \sum_{l \in path} m_{a,l}$$

2-3) If $end\_node(a) = D$, then set $UB = new\_cost$ and $P = path \cup \{a\}$ Continue

to the next arc.

2-4) Call *visit* (*end_node* (a) , $path \cup \{a\}$, *new_cost*).

3) Mark *vertex* as 'unvisited'.

Note that all variables except for the arguments of the procedure *visit* are global variables. The function *end_node* (a) returns the end node of directed arc a. Dijkstra's algorithm can be applied at step 1) in the procedure *main()* because c_a' is assumed to be non-negative. The procedure *visit* performs the depth-first search. *UB* provides the best upper bound of the shortest path length from S to D ever found. If a path is routed over arc a, at least c_a' is imposed on the path. Thus, $d(v)$ gives the lower bound of the cost from v to D. This lower bound is used to eliminate an unnecessary path search at step 2-2.

Appendix R. Initialization Procedures for the SVPR problems

The primal simplex algorithm requires an initial basic feasible solution. Although a feasible solution can be readily obtained for the SCFA-MF problems, a feasible commodity flow is not always apparent for the SVPR-MF problems. The two-phase approach for the simplex algorithm should be applied if a feasible solution is not immediately available [54]. This appendix gives the details on the initial procedures for the SVPR-MF-LINE problem and the SVPR-MF-ETE problem.

For each problem, the initial procedure takes the following steps:

Step 1. Obtain a shortest route commodity flow. If it is feasible, go to Step 3.

Step 2. Find an initial feasible commodity flow by solving an LP problem with artificial variables.

Step 3. Obtain an initial feasible solution to the SVPR problem.

Step 4. Restore a flow over the candidate restoration paths one by one.

The last step is not mandatory. However, this procedure helps to reduce a lost flow without incurring simplex iterations, which significantly shortens the computation time. Candidate restoration paths can be taken from the ones appearing in the optimal solution upon the network design (the solution to the SCFA-MF problems). If the speed-up version of the SVPR-MF solution procedures

is engaged, further candidate paths are generated from the beginning. See Section 4.2.4. and Section 4.3.4. for details.

R.1. Initialization procedure for SVPR-MF-LINE

If a shortest route flow is not feasible, then the following auxiliary linear programming problem must be solved to find an initial feasible basic solution (Step 2). Let $\hat{\mathbf{a}} = (\hat{a}_a)$ $(a \in A)$ be a vector of artificial variables where $\hat{a}_a$ is equal to the amount of violation on the capacity constraint for arc a.

Minimize $\sum_{a \in A} \hat{a}_a$

over $\underline{x} \geq \mathbf{0}$ and $\hat{\mathbf{a}} \geq \mathbf{0}$

subject to

$$\sum_{p \in P^\pi} x_p^\pi = q^\pi \qquad \forall \pi \in \Pi \qquad \textit{(R-1)}$$

$$\sum_{p \in P} \varphi_{a,p} \cdot x_p - \hat{a}_a \leq c_a \qquad \forall a \in A \qquad \textit{(R-2)}$$

Initial basic variables for this auxiliary problem consist of one commodity flow variable x_p per commodity (based on a shortest route flow), and either a slack variable s_a or an artificial variable $\hat{a}_a$ per arc (depending on the conformity to the capacity constraint). Note that we don't have to generate artificial variable $\hat{a}_a$ if $s_a \geq 0$ in the first place. Furthermore, once $\hat{a}_a$ becomes zero, it can be removed. The problem is infeasible if the optimum solution of the above problem is non-zero.

The solution procedure to this auxiliary problem is again decomposed into a master process and a sub-process. The master process checks to see if the current solution is globally optimal. Apparently, if all artificial variables are out of the basis, the solution is optimal and the procedure terminates. Otherwise, the following dual feasibility condition must be tested in order to find paths to generate:

$$-\sum_{a \in p} \mu_a \geq \sigma^{\pi} \; \forall p \in P^{\pi} \qquad \textit{(R-3)}$$

where σ^{π} and μ_a are the simplex multipliers corresponding to the constraints (R-1) and (R-2), respectively. An all-to-all shortest path algorithm [4] can be employed to find the shortest paths for all commodities under the arc cost $\{-\mu_a \geq 0\}$. Generate the shortest path variables violating the condition (R-3). If the condition is satisfied for all commodities, the solution is optimal for the auxiliary problem, implying that no feasible commodity flow exists.

The sub-process employs the revised simplex algorithm and finds an optimal solution using only the generated columns. Since the solution becomes optimal if all artificial variables become non-basic, the process attempts to select an artificial variable as a leaving variable during the course of simplex iterations. The basis matrix can be arranged and decomposed as follows, which could reduce the computation time of the simplex algorithm:

$$\begin{bmatrix} I & & C \\ A_1 & -I & M_1 \\ A_2 & I & M_2 \\ A_3 & & M_3 \end{bmatrix} = \begin{bmatrix} I & & \\ A_1 & I & \\ A_2 & & I \\ A_3 & & & L \end{bmatrix} \cdot \begin{bmatrix} I & & C \\ & -I & M_1 - A_1 C \\ & & I \quad M_2 - A_2 C \\ & & U \end{bmatrix}$$

where $LU = M_3 - A_3 C$. The first set of columns corresponds to key path flow variables, the second set to artificial variables, the third set to slack variables and the last set contains non-key path flow variables.

After a feasible commodity flow is found, we must build an initial feasible solution to the SVPR-MF-LINE problem (Step 3). The following procedures achieve this task.

- A key commodity flow variable remains a key with the same amount of flow.
- All non-key basic commodity flow variables also remain the same.
- All restoration flow variables are non-basic.
- All lost flow variables t_a ($\forall a \in A$) are in the basis, and the value is obtained by the restoration flow constraints.
- For a basic slack variable s_a, create basic slack variables s_a^l for $\forall l \in E_a$ with $s_a^l = s_a$.
- For a non-basic slack variable s_a, produce basic slack variables $s_a^l = 0$ for all $l \in E_a$ but one arbitrary $l' \in E_a$.

R.2. Initialization procedure for SVPR-ME-ETE

The initial procedure is the same as that for the SVPR-MF-LINE problem described above, except for Step 3. An initial feasible basic solution to the SVPR-MF-ETE problem can be settled as follows:

- First, generate all (π, l) rows pertaining to commodity flow variables generated in Step 1 and Step 2.
- A key commodity flow variable remains a key with the same amount of flow.
- All non-key basic commodity flow variables also remain the same.
- Lost flow variables $t^{\pi, l}$ for all generated (π, l) rows are in the basis, and the value is obtained from the restoration flow constraints. All other lost flow variables are undefined.
- All restoration flow variables are non-basic.
- Calculate a value of slack s_a^l for all capacity constraints.
- For each non-key basic commodity flow, pick up one slack variable with $s_a^l = 0$.
- Place all slack variables but the ones selected in the previous step in the basis.

Appendix S. Lagrangian Relaxation and Subgradient Method

This appendix briefly reviews the Lagrangian relaxation technique. It has been applied to a wide range of optimization problems in order to make them computationally tractable. This appendix also describes the subgradient method which has been successfully employed in the Lagrangian dual problem.

S.1. Lagrangian relaxation

The idea of *relaxation* is to replace the optimization problem by an easier problem. The relaxation technique can be used to obtain a lower bound in case of a minimization problem. Furthermore, it could give a way to produce a good approximation of the original problem. Now, a relaxation is defined as follows [63]. Consider the minimization problem *(P)*:

$(P) \quad z_P = \min \{ f(\mathbf{x}) : \mathbf{x} \in X \}$

Then, a minimization problem *(RP)* satisfying the conditions (S-1) and (S-2) is called a *relaxation* problem of *(P)*.

$$(RP)\quad z_{RP} = \min\{f_{RP}(\mathbf{x}) : \mathbf{x} \in X_{RP}\}$$

$$X_{RP} \supseteq X \qquad (S\text{-}1)$$

$$f_{RP}(\mathbf{x}) \leq f(\mathbf{x}) \text{ for } \forall \mathbf{x} \in X \qquad (S\text{-}2)$$

If *(RP)* is feasible, the following relation holds:

$$z_{RP} \leq z_P \qquad (S\text{-}3)$$

The simplest relaxation method is a *constraint relaxation* where some of the constraints are removed while $f_{RP}(\mathbf{x}) = f(\mathbf{x})$. The above two conditions are clearly satisfied.

A *Lagrangian relaxation* technique also removes constraints from the original problem. Typically complicated constraints are relaxed to simplify the problem. Then each relaxed constraint is added into the objective function after being multiplied by a distinct Lagrangian multiplier. Let $X = \{\mathbf{x} : \mathbf{x} \in X_{RP}, g_i(\mathbf{x}) \leq 0, i \in I\}$, where $g_i(\mathbf{x}) \leq 0, i \in I$ represents a complicated constraint to be relaxed. Then, apparently condition (S-1) holds. The following definition of $f_{RP}(\mathbf{x})$ also satisfies condition (S-2):

$$f_{RP}(\mathbf{x}) \equiv f(\mathbf{x}) + \sum_{i \in I} \mu_i \cdot g_i(\mathbf{x}) \text{ and } \mu_i \geq 0 \text{ for } \forall i \in I \qquad (S\text{-}4)$$

where $\mu = (\mu_i), i \in I$ is a Lagrangian multiplier. Define a *Lagrangian function* $\Im(\mu)$ as:

$$\Im(\mu) \equiv \min\{f_{RP}(\mathbf{x}) : \mathbf{x} \in X_{RP}\} \qquad (S\text{-}5)$$

This minimization problem is called as a *Lagrangian subproblem*. The Lagrangian function has been proven to be a non-smooth concave function of μ and is guaranteed to have a subgradient [60]. From (S-3), $\Im(\mu)$ is assured to be a lower bound of the problem *(P)* for any non-negative μ. The tightest lower bound can be found by maximizing the Lagrangian function over μ. This maximization problem leads to the following *Lagrangian dual problem*:

$$\Im^* \equiv \max\{\Im(\mu) : \mu \geq \mathbf{0}\} \qquad (S\text{-}6)$$

The subgradient algorithm described in the next section can be applied to the Lagrangian dual problem. By iteratively solving the Lagrangian subproblem and the Lagrangian dual problem, we

can refine the value of μ over iterations and obtain a tighter lower bound. The Lagrangian relaxation technique has been also employed to discover a good primal solution to problems with complicated constraints. Theoretically, it could eventually find an optimal solution if there is no duality gap. Although this may not be attained in practice, a solution of the Lagrangian subproblem could be utilized to find an approximate solution to the original problem [4]. Starting from a dually feasible solution, a primally feasible solution could be obtained with the aid of heuristics. The development of such heuristics and the selection of a set of constraints to be relaxed are two major design issues in the Lagrangian relaxation technique since they totally depend on the problem context.

S.2. Subgradient method

The subgradient method is developed for a concave (convex) non-differentiable optimization problem [74]. The scheme has been successfully applied to the Lagrangian dual problem [4] [63]. Now the Lagrangian relaxation problem can be solved by the following steps based on the subgradient method (Steps 2, 3 and 4):

0. Obtain an initial value of multiplier $\mu \geq \mathbf{0}$. $\mu = \mathbf{0}$ is typically used. Set $j = 1$.
1. For a given μ, solve the Lagrangian subproblem $\Im(\mu) \equiv \min\{f_{RP}(\mathbf{x}) : \mathbf{x} \in X_{RP}\}$. Suppose that $\mathbf{x}^*$ solves this problem.
2. Find any subgradient of $\Im(\mu)$ at $\mathbf{x}^*$. $(g_i(\mathbf{x}^*) : (i \in I))$ can be used [60].
3. Update a multiplier as follows:

$$\mu_i \leftarrow [\mu_i + \theta^j \cdot g_i(\mathbf{x}^*)]^+$$

where $[a]^+$ denotes a non-negative part of a, and θ^j is a step length at the j-th iteration.

4. Perform a termination test[22]. If not satisfied, then set $j \leftarrow j + 1$ and go back to Step 1.

Theoretically, the subgradient method converges if the step length θ^j is obtained from any sequence satisfying $\theta^j \to 0$ and $\sum \theta^j \to \infty$ [74]. For example, $\theta^j = 1/j$ satisfies this condition. However, several heuristics have been developed and widely applied in practice for step length selection in order to accelerate the convergence speed [4] [31] [39]. Based on our experiment, the heuristic developed by Held et. al. [39] turns out to have a good convergence property in the SVPR-KSP-LINE2 problem. We recite their method in the following. The step length is computed

22. For example, the procedure stops when a step length becomes sufficiently small.

by:

$$\theta^j = \frac{\lambda^j \cdot [UB - \Im(\mu)]}{\sum_{i \in I} (g_i(\mathbf{x}^*))^2} \qquad (S\text{-}7)$$

where *UB* is the best upper bound of the objective function found so far. λ^j is obtained as follows:

- $\lambda^j = 2$ for j up to $2n$, where n is a measure of the problem size. We make n equal to the number of constraints.
- Both λ^j and the number of iterations are halved until the number of iterations reaches some threshold value, z.
- Then λ^j is halved every z iterations until the resulting θ^j is sufficiently small.

Appendix T. Notations

This final appendix summarizes the notations used throughout the project.

T.1. General

$G = (V, A, \underset{\sim}{c})$	: A network.
V	: A set of vertices (nodes) representing ATM switches.
A	: A set of (directed) arcs representing optical trunks.
$\underset{\sim}{c} = (c_a) \quad a \in A$	: A vector of arc capacity.
E	: A set of (undirected) links. Each link is composed of two directed arcs with the same end-nodes but in opposite directions.
$E_a \equiv \{l \in E : a \notin l\}$	: A set of links excluding the one containing arc a.
Π	: A set of commodities. A commodity is a traffic flow from an origin to a destination. One commodity is defined for

each origin and destination pair.

P^{π} : A set of all possible routes for commodity $\pi \in \Pi$.

RP^{a} : A set of all possible or all candidate restoration routes for arc $a \in A$. (LINE related)

$RP^{\pi, l}$: A set of all possible or all candidate restoration routes for commodity $\pi \in \Pi$ against a failure of link $l \in E$. (ETE related)

$Q = (q^{\pi})\ \pi \in \Pi$: A vector of the requested bandwidth for each commodity.

$\underset{\sim}{f} = (f_a^{\pi})\ (\forall \pi \in \Pi, \forall a \in A)$: A commodity arc flow vector.

$f_a\ (\forall a \in A)$: The amount of an aggregate flow of arc a.

$$f_a = \sum_{\pi \in \Pi} f_a^{\pi}$$

$\underset{\sim}{x} = (x_p^{\pi})\ (\forall \pi \in \Pi, \forall p \in P^{\pi})$: A commodity path flow vector.

$\underset{\sim}{r} = (r_p^{a})\ (\forall a \in A, \forall p \in RP^{a})$: An arc restoration path flow vector. (LINE related)

$\underset{\sim}{r} = (r_p^{\pi, l})$: A restoration path flow vector. $\pi \in \Pi, l \in E, p \in RP^{\pi, l}$ (ETE related)

$r_a^l\ \forall a \in A, \forall l \in E_a$: The amount of bandwidth of arc a released by affected VP's due to a failure of link l. (ETE related)

$s_a^l\ (\forall a \in A, \forall l \in E_a)$: The slack variable of the capacity constraint for arc a upon a failure of link l.

$\theta_{a, p}$: An arc-path indicator variable which equals 1 if arc a is contained in path p, and 0 otherwise.

$\theta_{a, l, p}$: An indicator variable which equals 1 if both arc a and

link l are contained in path p, and 0 otherwise.

σ^{π} $(\forall \pi \in \Pi)$: The price of commodity (dual variable).

w_a $(\forall a \in A)$: The price of arc restoration (dual variable).

μ_a^l $(\forall a \in A, \forall l \in E_a)$: The price of arc per failure (dual variable).

$w^{\pi, l}$ $\forall \pi \in \Pi, \forall l \in E$: The price of link restoration per commodity (dual variable).

B : Basis matrix

$|X|$: Cardinality of a set X.

$[X] \equiv \{1, \ldots, |X|\}$

An (n, m) network : A network with n vertices and m arcs.

$<a, b>$: A commodity from a source node a to a destination node b.

a-b-c : A path from node a to c through b.

T.2. SCFA related

d_a $(\forall a \in A)$: A unit arc installation cost. It is assumed to be non-negative.

$\underline{z} = (z_a)$ $(\forall a \in A)$: A spare capacity vector.

z_a^l $(\forall a \in A, \forall l \in E_a)$: The amount of necessary spare bandwidth of arc a upon a failure of link l. (SCFA only)

T.3. SVPR related

$\underset{\sim}{t} = (t_a) \quad (\forall a \in A)$: An arc lost flow vector. (LINE related)

$\underset{\sim}{t} = (t^{\pi, l}) \quad \forall \pi \in \Pi, \forall l \in E$: A lost flow vector. (ETE related)

$z_a^l \; (\forall a \in A, \forall l \in E_a)$: The amount of total restoration flow of arc a upon a failure of link l. (SVPR only)

$\mu_a^l \quad (\forall a \in A, \forall l \in E)$: The price of arc per failure (dual variable). (ETE related)

$L(\underset{\sim}{f})$: An expected lost flow given a flow $\underset{\sim}{f}$.

$L_l(\underset{\sim}{f})$: A lost flow due to a failure of link l given a flow $\underset{\sim}{f}$.

$\phi_{v, a}$: A node-arc incidence variable which equals 1 if arc a leaves node v, -1 if arc a enters node v, and 0 otherwise.

η_v^{π} : An indicator variable which takes 1 if v is an originating node of commodity π, -1 if v is a destination node of π, and 0 otherwise.

$\zeta_{\beta}^{(l, k)}$: The residual capacity of arc β after KSP restoration using the first k restoration paths upon a failure of link l.

$RP^{(a, k)}$: The k-th restoration path for arc a. A set of arcs over the path.

h : The step size to numerically calculate one-sided partial derivatives. (MFD related)

$RP_h^{(a, k)}$: The h-th arc in the k-th restoration path for arc a.

$KSP^{(a, k)}$: The KSP constraint for arc a and its k-th restoration path for (SVPR-KSP-LINE2).

$\tilde{\rho} = (\rho_{a,k})$: Lagrangian multiplier for each KSP constraint.

$\Im_{\rho}(\underset{\sim}{x}, \underset{\sim}{r}, \underset{\sim}{t})$: Lagrangian function.

Bibliography

[1] CCITT recommendation, I series (B-ISDN), Nov. 1990.

[2] Special issue on "Integrity of public telecommunication networks," *IEEE JSAC*, vol.12, no.1, Jan. 1994.

[3] Special section on "Network reliability - The state of the art," *IEEE Trans. on Reliability*, vol.35, no.3, Aug. 1986.

[4] R.K.Ahuja, T.L.Magnanti and J.B.Orlin, "Network flows: Theory, algorithms, and applications," Prentice Hall, Englewood Cliffs, N.J., 1993.

[5] J.Anderson, B.T.Doshi, S.Dravida and P.Harshavardhana, "Fast restoration of ATM networks," *IEEE JSAC*, vol.12, no.1, pp.128-138, Jan. 1994.

[6] T.Aoyama, I.Tokizawa and K.Sato, "ATM VP-based broadband networks for multimedia services," *IEEE Comm. Magazine*, vol.31, no.4, pp.30-39, Apr.1993.

[7] G.R.Ash, R.H.Cardwell and R.P.Murray, "Design and optimization of networks with dynamic routing," *Bell System Technical Journal*, vol.60, no.8, pp.1787-1820, Oct. 1981.

[8] G.R.Ash, "Dynamic network evolution, with examples from AT&T's evolving dynamic network," *IEEE Comm. Magazine*, vol.33, no.7, pp.26-39, Jul. 1995.

[9] A.A.Assad, "Multicommodity network flows - A survey", *Networks*, vol.8, pp.31-91, 1978.

[10] J.J.Bae and T.Suda, "Survey of traffic control schemes and protocols in ATM networks," *Proceedings of the IEEE*, vol.79, no.2, pp.170-189, Feb. 1991.

[11] J.E.Baker, "A distributed link restoration algorithm with robust preplanning," *IEEE Globecom '91*, pp.306-311, Dec. 1991.

[12] R.H.Bartels and G.H.Golub, "The simplex method of linear programming using LU decomposition," *Comm. ACM*, vol.12, no.5, pp.266-268, May 1969.

[13] D.Bertsekas and R.Gallager, "Data networks," Prentice Hall Inc., Englewood Cliffs, N.J., 1987.

[14] D.Bertsekas, "Linear network optimization: Algorithms and codes," MIT Press, 1991.

[15] J.Bicknell, C.E.Chow and S.Syed, "Performance analysis of fast distributed network restoration algorithms," *IEEE Globecom '93*, pp1596-1600, Nov. 1993.

[16] J.A.Bondy and U.S.R.Murty, "Graph theory with applications," McGraw-Hill, 1976.

[17] V.Chvatal, "Linear programming," W.H.Freeman, New York, N.Y., 1983.

[18] B.A.Coan, M.P.Vecchi and L.T.Wu, "A Distributed protocol to improve the survivability of trunk networks," *13th International Switching Symposium (ISS '90)*, pp.173-179, May 1990.

[19] R.Doverspike, "A multi-layered model for survivability in intra-LATA transport networks," *IEEE Globecom '91*, pp.2025-2031, Dec. 1991.

[20] D.A.Dumm, W.D.Grover and M.H.MacGregor, "Comparison of k-shortest paths and maximum flow routing for network facility restoration," *IEEE JSAC*, vol.12, no.1, pp.88-99, Jan. 1994.

[21] A.I.Elwalid and D.Mitra, "Effective bandwidth of general markovian traffic sources and admission control of high speed networks (Extended abstract)," *IEEE INFOCOM '93*, pp.256-265, Mar. 1993.

[22] L.Fratta, M.Gerla and L.Kleinrock, "The flow deviation method: An approach to store-and-forward communication network design," *Networks*, vol.3, pp.97-133, 1973.

[23] H.Fujii and N.Yoshikai, "Restoration message transfer mechanism and restoration characteristics of double-search self-healing ATM networks," *IEEE JSAC*, vol.12, no.1, pp.149-158, Jan. 1994.

[24] B.Gavish and I.Neuman, "A system for routing and capacity assignment in computer communication networks," *IEEE Trans. on Comm.*, vol.37, no.4, pp.360-366, Apr. 1989.

[25] B.Gavish and I.Neuman, "Routing in a network with unreliable components," *IEEE Trans. on Comm.*, vol.40, no.7, Jul. 1992.

[26] M.Gerla, "The design of store-and-forward networks for computer communications," Ph.D. Dissertation, Computer Science Department, University of California, Los Angeles, Jan. 1973.

[27] M.Gerla, J.A.S.Monteiro and R.Pazos, "Topology design and bandwidth allocation in ATM nets," *IEEE JSAC*, vol.7, no.8, pp.1253-1262, Oct. 1989.

[28] A.Gersht and A.Shulman, "Optimal routing in circuit switched network," *IEEE Trans. on Comm.*, vol.37, no.11, pp.1203-1211, Nov. 1989.

[29] A.Gersht and S.Kheradpir, "Real-time bandwidth allocation and path restorations in SONET-based self-healing mesh networks," *IEEE ICC '93*, pp.250-255, May 1993.

[30] A.Gersht and A.Shulman, "Optimal dynamic virtual path bandwidth allocation and restoration in ATM networks," *IEEE Globecom '94*, pp.770-776, Nov. 1994.

[31] J.L.Goffin, "On convergence rates of subgradient optimization methods," *Mathematical programming*, no.13, pp.329-347, 1977.

[32] B.L.Golden, "Point estimation of a global optimum for large combinatorial problems," *Communications in Statistics, Simulation and Computation*, vol.B7, no.4, pp.361-367, 1978.

[33] B.L.Golden and F.B.Alt, "Interval estimation of a global optimum for large combinatorial problems," *Naval Research Logistics Quarterly*, vol.26, pp.69-77, 1979.

[34] P.E.Green Jr., "Fiber optic networks," Prentice Hall Inc., Englewood Cliffs, N.J., 1993.

[35] W.D.Grover, "The selfhealing network: A fast distributed restoration technique for networks using digital cross-connect machines," *IEEE Globecom '87*, pp.1090-1095, Dec. 1987.

[36] W.D.Grover, B.D.Venables, M.H.MacGregor and J.H.Sandham, "Development and performance assessment of a distributed asynchronous protocol for real-time network restoration," *IEEE JSAC*, vol.9, no.1, pp.112-125, Jan. 1991.

[37] W.D.Grover, T.D.Bilodeau and B.D.Venables, "Near optimal spare capacity planning in a mesh restorable network," *IEEE Globecom '91*, pp.2007-2012, Dec. 1991.

[38] R.Guerin, H.Ahmadi and M.Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE JSAC*, vol.9, no.7, pp.968-981, Sep. 1991.

[39] M.Held, P.Wolfe and H.P.Crowder, "Validation of subgradient optimization," *Mathematical programming*, no.6, pp.62-88, 1974.

[40] J.Y.Hui, M.B.Gursoy, N.Moayeri and R.D.Yates, "A layered broadband switching architecture with physical and virtual path configurations," *IEEE JSAC*, vol.9, no.9, pp.1416-1426, Dec. 1991.

[41] C.G.Kang and H.H.Tan, "Fault-tolerant capacity and flow assignment in packet switched networks," *IEEE MILCOM '92*, pp.165-171, Oct. 1992.

[42] N.Karmarkar, "A new polynomial time algorithm for linear programming," *Combinatorica*, vol.4, pp.373-395, 1984.

[43] R.Kawamura, K.Sato, and I.Tokizawa, "Self-healing ATM networks based on virtual path concept," *IEEE JSAC*, vol.12, no.1, pp.120-127, Jan. 1994.

[44] L.Kleinrock, "Analytic and simulation methods in computer network design," *Spring Joint Comput. Conf.*, AFIPS Conf. Proc., vol.36, pp.568-579, 1970.

[45] H.Komine, T.Chujo, T.Ogura, K.Miyazaki and T.Soejima, "A distributed restoration algorithm for multiple link and node failures of transport networks," *IEEE Globecom '90*, pp.459-463, Dec. 1990.

[46] K.R.Krishnan, "Dynamic traffic routing and network management," *IEEE Globecom '91*, pp.1346-1350, Dec. 1991.

[47] M-J.Lee and J.R.Yee, "A design algorithm for reconfigurable ATM networks," *IEEE INFOCOM '93*, pp.144-151, Mar. 1993.

[48] M-J.Lee and J.R.Yee, "Optimal minimax routing in ATM networks," *IEEE Globecom '90*, pp.869-873, Dec. 1990.

[49] C.Lemarechal, "Nondifferentiable optimization," In *Optimization* (G.L.Nemhauser et.al., Ed.) Handbooks in operations research and management science, vol.1, North Holland, pp.529-572, 1989.

[50] T.Leventhal, G.Nemhauser and L.Trotter, "A column generation algorithm for optimal traffic assignment," *Transportation Science*, vol.7, no.2, pp.168-176, May 1973.

[51] F.Y.S.Lin and K.T.Cheng, "Virtual path assignment and virtual circuit routing in ATM networks," *IEEE Globecom '93*, pp.436-441, Nov. 1993.

[52] N.D.Lin, A.Zolfaghari and B.Lusignan, ATM virtual path self-healing based on a new path restoration protocol," *IEEE Globecom '94*, pp.794-798, Nov. 1994.

[53] M.Logothetis, S.Shioda and G.Kokkinakis, "Optimal virtual path bandwidth management assuring network reliability," *IEEE ICC '93*, pp.30-36, May 1993.

[54] D.G.Luenberger, "Linear and nonlinear programming," 2nd ed., Addison-Wesley, Reading, M.A., 1984.

[55] J.C.McDonald, "Public network integrity - Avoiding a crisis in trust," *IEEE JSAC*, vol.12, no.1, pp.5-12, Jan. 1994.

[56] J.A.McEachern, "Gigabit networking on the public transmission network," *IEEE Comm.Mag.*, vol.30, no.4, pp.70-78, Apr. 1992.

[57] D.Medhi, "A unified framework for survivable telecommunications network design," *IEEE ICC '92*, pp.411-415, Jun. 1992.

[58] M.Minoux, "Optimum synthesis of a network with non-simultaneous multicommodity flow requirements," In *Studies on Graphs and Discrete Programming* (P.Hansen, Ed.) Annals of Discrete Mathematics 11, North Holland, pp.269-277, 1981.

[59] M.Minoux and J.Y.Serreault, "Subgradient optimization and large scale programming: An application to optimum multicommodity network synthesis with security constraints," *RAIRO / Operations Research*, vol.15, no.2, pp.185-203, May 1981.

[60] M.Minoux, "Mathematical programming: Theory and algorithm," John Wiley & Sons, 1986 (translated by S.Vajda).

[61] S.E.Minzer, "Broadband ISDN and asynchronous transfer mode (ATM)," *IEEE Comm. Mag.*, vol.27, no.9, pp.17-24, Sep. 1989.

[62] J.A.S.Monteiro and M.Gerla, "Topological reconfiguration of ATM networks," *IEEE INFOCOM '90*, pp.207-214, May 1990.

[63] G.L.Nemhauser and L.A.Wolsey, "Integer programming," In *Optimization* (G.L.Nemhauser et.al., Ed.) Handbooks in operations research and management science, vol.1, North Holland, pp.447-527, 1989.

[64] S.Ohta, K.Sato and I.Tokizawa, "Dynamically controllable ATM transport network based on the virtual path concept," *IEEE Globecom '88*, pp.1272-1276, Dec.1988.

[65] S.Ohta and K.Sato, "Dynamic bandwidth control of the virtual path in an asynchronous transfer mode network," *IEEE Trans. on Comm.*, vol.40, no.7, pp.1239-1247, Jul. 1992.

[66] R.Ramaswami, "Multiwavelength lightwave networks for computer communication," *IEEE Comm. Mag.*, vol.31, no.2, pp.78-88, Feb. 1993.

[67] J.K.Reid, "A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming basis," *Mathematical Programming*, vol.24, pp.55-69, 1982.

[68] H.Sakauchi, Y.Nishimura and S.Hasegawa, "A self-healing network with an economical spare-channel assignment," *IEEE Globecom '90*, pp.438-443, Dec. 1990.

[69] H.Sakauchi, Y.Okanoue and S.Hasegawa, "Spare-channel design schemes for self-healing networks," *IEICE Trans.*, vol.E75-B, no.7, pp.624-633, Jul. 1992.

[70] B.Sanso, F.Soumis and M.Gendreau, "On the evaluation of telecommunications network reliability using routing models," *IEEE Trans. on Comm.*, vol.39, no.10, pp.1494-1501, Oct. 1991.

[71] K.Sato, S.Ohta and I.Tokizawa, "Broad-band ATM network architecture based on virtual paths," *IEEE Trans. on Comm.*, vol.38, no.8, pp.1212-1222, Aug. 1990.

[72] K.Sato and I.Tokizawa, "Flexible asynchronous transfer mode networks utilizing virtual paths," *IEEE ICC '90*, pp.831-838, Apr. 1990.

[73] S.Shioda and H.Uose, "Virtual path bandwidth control method for ATM networks: Successive modification method," *IEICE Transactions*, vol.E74, no.12, pp.4061-4068, Dec.1991.

[74] N.Z.Shor, "Minimization methods for non-differentiable functions," translated by K.C.Kiwiel and A.Ruszczynski, *Springer series in computational mathematics* 3, Springer-Verlag, Berlin, Germany, 1985.

[75] J.Sosnosky, "Service applications for SONET DCS distributed restoration," *IEEE JSAC*, vol.12, no.1, pp.59-68, Jan. 1994.

[76] H.Tatsuno and N.Tokura, "Hitless path protection switching techniques for ATM networks," *Trans. on IEICE*, vol.J76B-I, no.6, pp.421-430, Jun.1993. *(in Japanese)*

[77] A.Torn and A.Zilinskas, "Global optimization," In *Lecture Notes in Computer Science* 350 (G.Goos and J.Hartmanis Ed.), Springer-Verlag, Berlin, Germany, 1987,

[78] B.D.Venerables, W.D.Grover and M.H.MacGregor, "Two strategies for spare capacity placement in mesh restorable networks," *IEEE ICC '93*, pp.267-271, May 1993.

[79] J.S.Whalen and J.Kenney, "Finding maximal link disjoint paths in multigraph," *IEEE Globecom '90*, pp.470-474, Dec. 1990.

[80] C.H.Yang and S.Hasegawa, "FITNESS: Failure immunization technology for network service survivability," *IEEE Globecom '88*, pp.1549-1554, Dec. 1988.

[81] W.L.Winston, "Operations research: Applications and algorithms," PWS-KENT Publishing Co., Boston, M.A., 1987.

[82] T-H.Wu, "Fiber network service survivability," Artech House, Norwood, M.A., 1992.

[83] T-H.Wu, H.Kobrinski, D.Ghosal, and T.V.Lakshman, "The impact of SONET digital cross-connect system architecture on distributed restoration," *IEEE JSAC*, vol.12, no.1, pp.79-87, Jan. 1994.

*U.S. GOVERNMENT PRINTING OFFICE: 1997-509-127-47129

MISSION OF ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.